Contents

[Matrix Formula – Công Thức Ma Trận: 1](#_Toc157011407)

[Square Matrix – Ma Trận Vuông: 1](#_Toc157011408)

[Animation Matrix – Ma Trận Hoạt Ảnh: 2](#_Toc157011409)

[Linear Transformation – Biến Đổi Tuyến Tính: 4](#_Toc157011410)

[Special Matrix – Ma Trận Đặc Biệt: 11](#_Toc157011411)

[Matrix Form – Dạng Của Ma Trận: 15](#_Toc157011412)

[Complex Matrix – Ma Trận Phức: 16](#_Toc157011413)

[Vector Space – Không Gian Vector: 17](#_Toc157011414)

[Product – Tích: 21](#_Toc157011415)

[Gaussian Elimination – Phép Khử Gauss: 23](#_Toc157011416)

[Model – Mô Hình: 26](#_Toc157011417)

Matrix Formula – Công Thức Ma Trận:

Square Matrix – Ma Trận Vuông:

1. Cấp Của Ma Trận Vuông?

* Là số hàng của nó

1. Vết (Trace) Của Ma Trận Vuông?

* Là tổng các giá trị trên đường chéo chính của ma trận vuông
* Ví dụ

1. Định Thức Con (Minor) Và Bù Đại Số (Cofactor) Của Ma Trận Vuông?

* Xét phần tử ở hàng i cột j, tính từ trên xuống, trái sang, bắt đầu từ 1 của ma trận vuông A
* Phần tử này sẽ tương ứng với định thức con có giá trị bằng định thức của cái ma trận sau khi lấy A bỏ đi hàng i cột j, kí hiệu là Mij
* Nếu i + j chẵn, thì bù đại số của phần tử = định thức con của nó, nếu lẻ, thì = âm định thức con của nó, kí hiệu là Aịj

1. Cách Tính Định Thức Ma Trận Vuông Bằng Mở Rộng Laplace?

* Cho ma trận A, chọn 1 dòng hay cột bất kì, lấy mỗi phần tử trên đó nhân với bù đại số tương ứng rồi cộng tất cả lại được định thức của A

1. Ma Trận Phụ Hợp (Cofactor Matrix)?

* Cho ma trận A, ma trận phụ hợp của A được tạo ra bằng cách thay mỗi phần tử của A bằng bù đại số của nó
* Ví dụ

1. Ma Trận Liên Hợp (Adjugate Matrix, Adjoint Matrix)?

* Là ma trận chuyển vị của ma trận phụ hợp
* Ví dụ

1. Tại Sao Gọi Là Ma Trận Liên Hợp?

* Giả sử ta nhân số phức A cho liên hợp của nó, sẽ ra số thực có độ lớn = bình phương Module của A, giả sử A là ma trận, thì khi A nhân cho ma trận liên hợp của nó, sẽ được ma trận đường chéo vuông mà mọi phần tử trên đường chéo chính đều = định thức của A
* Ví dụ

1. Tính Ma Trận Nghịch Đảo Của Ma Trận Vuông?

1. Kí Hiệu Ma Trận Lũy Thừa Nghĩa Là Gì?

* An nghĩa là ma trận vuông A nhân cho chính nó n lần
* A0 = I

Animation Matrix – Ma Trận Hoạt Ảnh:

1. Ma Trận Biến Hình Thành Hình Đối Xứng Của Nó Qua Gốc Tọa Độ?

1. Ma Trận Quay Quanh Trục Cao, Hướng Ngược Chiều Kim Đồng Hồ Khi Nhìn Từ Trên Xuống?

* θ là góc quay

1. Ma Trận Scale, Quay Rồi Mới Tịnh Tiến Trong Không Gian 3D?

* Lưu ý phải đổi tọa độ điểm từ 3 chiều sang 4 chiều, 3 phần tử đầu giá trị giữ nguyên, phần tử thứ 4 giá trị luôn = 1
* Trong đó, ma trận Scale rồi quay là

* Vector tịnh tiến là

* Ví dụ dạng 4 chiều của tọa độ điểm

1. Ma Trận Shear?

* Shear theo trục hoành

* Shear theo trục tung

* Ma trận Identity cũng là ma trận Shear với a = 0

1. Ma Trận Chiếu Vuông Góc 1 Không Gian Vector Vào 1 Subspace Của Nó?

* A là ma trận m x n, m > n, có không gian cột là Subspace đang nói đến, khi đó ma trận chiếu là

* P có kích thước m x m
* Ví dụ
* Ta muốn tạo ra 1 ma trận P có tác dụng ánh xạ 1 điểm trong không gian 3D vào hình chiếu của nó trên 1 mặt phẳng qua gốc tọa độ, mặt phẳng này chính là 1 Subspace của không gian 3D, gọi A là ma trận có không gian cột là mặt phẳng này, giả sử

* Chứng minh công thức trên, đầu tiên P phải ánh xạ A vào chính nó

* Tiếp theo, Null Space của P phải vuông góc không gian cột của P, vâng, vì các Vector đơn vị bị ánh xạ lên A, nên không gian cột của P cũng chính là A, có như vậy thì nó mới chiếu vuông góc, mà ta lại có

* Như vậy P đối xứng, hay không gian hàng = không gian cột, mà Null Space vuông góc không gian hàng, nên nó cũng vuông khóc không gian cột

Linear Transformation – Biến Đổi Tuyến Tính:

1. Diện Tích Của 1 Hình Khi Bị Biến Đổi Bởi 1 Ma Trận?

* Diện tích của nó sẽ nhân k lần, k là định thức của ma trận

1. Định Thức Ma Trận Có Tỉ Lệ Với Độ Tăng Chiều Dài?

* Không, chiều dài của 1 đoạn thẳng bị biến đổi với ma trận không tỉ lệ với định thức của ma trận biến đổi nó

1. Biến Đổi Affine?

* Là biến đổi tuyến tính rồi tịnh tiến

1. Biến Đổi Cứng (Rigid Transformation, Euclidean Transformation)?

* Là 1 tổ hợp các phép quay, tịnh tiến, và Flip

1. Không Gian Hàng Và Không Gian Cột?

* Xét ma trận A có kích thước m x n
* Không gian hàng là Span của tất cả các hàng, kí hiệu C(AT)
* Không gian cột là Span của tất cả các cột, kí hiệu C(A)
* Số chiều của không gian hàng và không gian cột luôn = nhau và = hạng của ma trận = r(A)
* Thêm 1 cột hoặc 1 hàng mới vào A, thì hạng của A chỉ có thể tăng 1 hoặc giữ nguyên

1. Range Space?

* Range Space là không gian ảnh của 1 không gian V nào đó bị biến đổi bởi 1 ánh xạ Homomorphism f, còn gọi là ảnh của ánh xạ f
* Số chiều của Range Space gọi là hạng của ánh xạ f
* Kí hiệu

1. Null Space Và Left Null Space?

* Null Space là Kernel của Homomorphism f ánh xạ từ A tới B
* Số chiều của Null Space gọi là số vô hiệu của ánh xạ f (Nullity)
* Kí hiệu

* e là Identity của B
* Left Null Space của 1 ma trận là Null Space của ma trận chuyển vị của ma trận đó
* Null Space của 1 ma trận luôn vuông góc với không gian hàng của ma trận đó, tương tự Left Null Space luôn vuông góc không gian cột

1. Tính Số Chiều Của Null Space Và Left Null Space Của 1 Ma Trận?

* Số chiều của Null Space là

* r là hạng của ma trận
* n là số cột của ma trận
* Số chiều của Left Null Space là

* m là số hàng của ma trận

1. Eigen Value Và Eigen Vector?

* Lưu ý ở đây ta đang xét trong không gian phức
* Cho ma trận vuông A m x m, khi áp dụng ma trận này để biến đổi không gian, thì để ý thấy, có những Vector khác Vector 0 chỉ bị Scale chứ không bị quay, đây gọi là những Eigen Vector của A, hệ số Scale gọi là Eigen Value
* Mỗi Eigen Value sẽ có vô hạn Eigen Vector ứng với nó, tuy nhiên A chỉ có đúng m Eigen Vector độc lập tuyến tính, tính cả trường hợp đáng lẽ độc lập tuyến tính nhưng bị cùng phương
* Từ điều trên, dễ thấy A sẽ có đúng m Eigen Value khác nhau ứng với m Eigen Vector độc lập tuyến tính, trừ trường hợp 2 Eigen Vector đáng lẽ độc lập tuyến tính nhưng bị cùng phương thì chắc chắn có chung 1 Eigen Value
* Eigen Vector đáng lẽ độc lập tuyến tính như bị cùng phương được hiểu như sau, ban đầu có 2 Eigen Vector độc lập tuyến tính là X và Y, Eigen Value khác nhau, bằng phép nội suy, bạn dần biến phương của X giống phương của Y, độ dài Y giữ nguyên, độ dài X tăng dần tới vô cực, đồng thời dần biến Eigen Value của X giống Eigen Value của Y, khi làm như vậy, ma trận ban đầu sẽ dần biến thành ma trận Shear, X và Y hợp thành 1, nhưng nếu độ dài X không tăng dần tới vô cực mà giữ nguyên thì ma trận Shear ở đây chính là ma trận Identity luôn, X và Y vẫn là 2 bản thể độc lập tuyến tính riêng biệt và bất kì cặp Vector độc lập tuyến tính nào trong Span của X, Y lúc ban đầu cũng có thể trở thành cặp Eigen Vector X, Y hiện tại
* Eigen Space của A là Span của tất cả Eigen Vector độc lập tuyến tính của A
* Bản chất của phép biến đổi tuyến tính A đó là Scale theo Basis là Eigen Vector với hệ số Scale là Eigen Value

1. Tính Eigen Value Và Eigen Vector?

* λ là Eigen Value khi và chỉ khi nó thỏa mãn phương trình đặc trưng sau

* A là ma trận cần tính Eigen Value và Eigen Vector
* I là ma trận Identity
* Giả sử λ là nghiệm kép, hoặc bội 3, …, hay tổng quát là bội n của phương trình trên, thì n gọi là bội đại số (Algebraic Multiplicity) của λ
* Với mỗi giá trị của λ, giải phương trình sau là ra Eigen Vector v tương ứng

* Nói cách khác, tập hợp các Vector v thỏa mãn phương trình trên chính là Null Space của ma trận A – λI, còn gọi là không gian con riêng của λ, kí hiệu Eλ, số chiều của Eλ gọi là bội hình học (Geometric Multiplicity) của λ
* Bội hình học ≤ bội đại số
* Đa thức đặc trưng của A là

1. Eigen Decomposition?

* Là phân tích 1 ma trận vuông A m x m, A có đúng m Eigen Vector độc lập tuyến tính, thành tích của Eigen Vector và Eigen Value

* Q là ma trận vuông m x m có các cột là Eigen Vector, gọi là ma trận chuyển đổi Mod (Modal Matrix) của A, độ dài mỗi Eigen Vector là bất kì
* Λ là ma trận đường chéo vuông m x m, mỗi phần tử trên đường chéo chính là Eigen Value
* Những ma trận mà có thể phân tích được kiểu này gọi là ma trận khả chéo hóa (Diagonalizable Matrix), nếu Q là ma trận trực giao thì gọi là ma trận khả chéo hóa trực giao
* Ma trận khả chéo hóa khi và chỉ khi bội đại số của mọi Eigen Value = bội hình học của nó
* Để chứng minh công thức trên, ta tưởng tượng 1 phép biến đổi tuyến tính bởi ma trận A lên 1 hình bình hành như sau



* Dễ thấy 2 mũi tên màu đỏ là 2 Eigen Vector độc lập tuyến tính nào đó của A, gọi ma trận có 2 cột là 2 Vector này là Q, cũng dễ thấy Eigen Value của mũi tên bên trái = 1.5, của mũi tên bên phải = 2
* Ta sẽ phân tích A thành nhiều phép biến đổi, ban đầu cho Q-1 tác dụng vào hình bình hành, do Q-1Q = I, kết quả ta được hình vuông đơn vị



* Tiếp theo Scale hình vuông này theo chiều dọc 1.5 lần và chiều ngang 2 lần, dễ thấy đây chính là các Eigen Value



* Cuối cùng dùng ma trận Q để đưa hình chữ nhật về hình dạng hình bình hành ban đầu nhưng 2 cạnh bị Scale bởi Eigen Value



* Vậy, phép biến đổi A thực chất cũng chỉ là ta sử dụng phép đổi Q-1 rồi sau đó dùng Λ và cuối cùng dùng Q
* Từ sơ đồ trên dễ thấy, định thức của A cũng chính bằng tích của toàn bộ Eigen Value

1. Ma Trận Đồng Dạng (Matrix Similarity)?

* 2 ma trận vuông A và B gọi là đồng dạng nếu tồn tại ma trận vuông P khả nghịch sao cho

* Điều này nghĩa là A và B có Eigen Value giống nhau, hay nói cách khác, bản chất của A và B là một, chỉ là chúng tác dụng lên Basis khác nhau

1. Spectral Decomposition?

* Là Eigen Decomposition với ma trận chuẩn

1. SVD (Singular Value Decomposition)?

* Mọi phép biến đổi tuyến tính đều có bản chất là quay và Scale, do đó, mọi ma trận đều có thể phân tích thành các ma trận quay và Scale, gọi ma trận cần phân tích là A, ta có

* Σ là ma trận đường chéo chữ nhật, chứa Singular Value trên đường chéo chính, vừa là nhân tố Scale, vừa là nhân tố giảm chiều hoặc tăng chiều
* U và V đều là ma trận trực giao, là nhân tố quay và Flip
* SVD giải quyết được bài toán là tìm hệ Vector vuông góc với nhau từng đôi một mà khi chịu tác dụng của A thì chúng vẫn vuông góc với nhau, vâng, hệ Vector đó là V, khi lấy A tác dụng lên V, được UΣ cũng là hệ Vector vuông góc
* Singular Value có thể được hiểu như sau, đó là phép biến đổi tuyến tính A thực chất là ta chọn 1 Basis là 1 ma trận vuông góc, và chỉ cần 1 lần Scale theo Basis đó là tương đương với A, khi đó hệ số Scale chính là Singular Value
* Tưởng tượng ta có phép biến đổi tuyến tính là ma trận A tác dụng lên hình vuông đơn vị



* Bây giờ ta sẽ phân tích A thành nhiều phép biến đổi, ban đầu cho V-1 tác dụng vào hình vuông đơn vị, tương đương với việc quay và Flip hình vuông đó



* Tiếp theo Scale hình vuông này theo chiều dọc 2 lần và chiều ngang 3 lần, dễ thấy đây chính là các Singular Value



* Cuối cùng, dùng U để quay vào đúng vị trí



* Vậy, phép biến đổi A thực chất cũng chỉ là quay với Scale

1. Tìm SVD Của 1 Ma Trận?

* Để tìm V, trước tiên tìm Eigen Decomposition của ATA, đây là ma trận đối xứng nên các Eigen Vector sẽ vuông góc với nhau, chọn các Vector có độ dài = 1 được V, đồng thời ta cũng tìm được ma trận đường chéo chứa Eigen Value của ATA là ΣTΣ, từ ΣTΣ tìm Σ, lưu ý chỉnh thứ tự các Eigen Value trong ΣTΣ sao cho mấy cái = 0 xuống hết phía dưới, đồng thời cũng chỉnh thứ tự các Eigen Vector trong V để cho đúng với Eigen Value
* Chứng minh điều trên

* Để tính U, ta suy ra từ công thức SVD

* Ta cũng có thể tính U trước bằng cách tìm Eigen Decomposition của AAT

* Rồi tính V bằng công thức

* Ví dụ

* Vậy

1. Tìm Nghịch Đảo Của Ma Trận Chữ Nhật?

* Cho ma trận chữ nhật A m x n, nếu m > n, nghịch đảo của A sẽ là ma trận A-1

n x m thỏa mãn

* Chứng minh điều trên, đầu tiên, tích A-1A phải là ma trận Identity

* Ngược lại, tích AA-1 có thể không phải ma trận Identity, do đó, A-1 còn được gọi là nghịch đảo bên trái của A
* Nếu m < n, nghịch đảo của A sẽ là ma trận A-1 n x m thỏa mãn

* Chứng minh điều trên, đầu tiên, tích AB phải là ma trận Identity

* Ngược lại, tích A-1A có thể không phải ma trận Identity, do đó, A-1 còn được gọi là nghịch đảo bên phải của A
* Ví dụ

* Bất kì ma trận chữ nhật không vuông nào cũng chỉ có nhiều nhất 1 nghịch đảo 1 bên, ví dụ ma trận A 5 x 3 có hạng = 3, A-1 là nghịch đảo bên trái của A, khi đó

ma trận AA-1 sẽ có kích thước 5 x 5, trong khi dễ thấy các cột của nó là tổ hợp tuyến tính của 3 cột trong A, do đó chỉ có hạng = 3, mà ma trận Identity 5 x 5 có hạng = 5, mâu thuẫn

1. Ma Trận Ánh Xạ Tuyến Tính Ứng Với 1 Cặp Cơ Sở?

* Cho cơ sở chính tắc A, và 2 cơ sở khác là B và C, cho M là ma trận nào đó biến đổi A
* Cho D là 1 Vector ứng với A, DB là biểu diễn của D trong B, khi cho M tác dụng lên A, D sẽ ánh xạ tới điểm D’ = MD, biểu diễn của D’ trong C là D’C
* Bây giờ ta muốn tìm 1 ma trận, sao cho nó biến DB thành D’C, vâng, 1 ma trận N sao cho NDB = D’C, ma trận N gọi là ma trận của ánh xạ tuyến tính M ứng với cặp cơ sở B và C
* Để tìm N
* Bước 1, tính MB, ta tính được sau khi bị M tác dụng thì B sẽ thành cái gì
* Bước 2, tìm biểu diễn của MB ứng với cơ sở C, được N
* Công thức tổng quát

Special Matrix – Ma Trận Đặc Biệt:

1. Ma Trận Không?

* Là ma trận sao cho toàn bộ phần tử = 0

1. Ma Trận Đường Chéo (Diagonal Matrix)?

* Là ma trận mà mọi phần tử không nằm trên đường chéo chính đều = 0
* Đường chéo chính là đường chéo đi qua góc trái trên
* Ví dụ

1. Ma Trận Mở Rộng (Augmented Matrix)?

* Là ma trận nhận được khi ghép 2 ma trận cùng số dòng lại với nhau theo chiều ngang
* Ví dụ

* Ma trận mở rộng của A và B là

1. Ma Trận Suy Biến (Singular Matrix)?

* Là ma trận vuông có định thức = 0
* Không khả nghịch
* Hệ phương trình thuần nhất có các hệ số của ma trận suy biến thì sẽ có vô số nghiệm
* Ví dụ

1. Ma Trận Không Suy Biến (Non Singular Matrix)?

* Là ma trận vuông có định thức khác 0
* Khả nghịch
* Hệ phương trình thuần nhất có các hệ số của ma trận không suy biến thì sẽ có nghiệm duy nhất là x = y = z = … = 0
* Ví dụ

1. Ma Trận Tam Giác Trên (Upper Triangular Matrix)?

* Là ma trận vuông mà mọi phần tử ở phía dưới đường chéo chính đều = 0
* Ví dụ

* Nếu mọi phần tử ở phía trên đường chéo chính đều = 0 thì gọi là ma trận tam giác dưới
* Tích 2 ma trận tam giác trên cũng là 1 ma trận tam giác trên, tương tự ma trận tam giác dưới
* Tất cả phần tử trên đường chéo chính của ma trận tam giác bất kì đều là Eigen Value của nó, do đó định thức của nó = tích các phần tử trên đường chéo chính

1. Ma Trận Sơ Cấp (Elementary Matrix)?

* Là ma trận E đạt được = cách áp dụng 1 lần phép biến đổi Gauss bất kì trên ma trận Identity
* Ví dụ

* Thực hiện nhân hàng 2 với 3, ta được 1 ma trận sơ cấp

* Cho ma trận A kích thước m x n, khi này giả sử ta thực hiện 1 lần phép biến đổi Gauss nào đó trên 1 hàng nào đó của A, thì kết quả tương đương với việc lấy 1 ma trận sơ cấp m x m ứng với phép biến đổi Gauss đó nhân với A
* Ví dụ

* Sử dụng phép biến đổi Gauss thay hàng 2 = hàng 3 + 0.5 \* hàng 2, ta được

* Cũng lấy ma trận I 3 x 3, và thực hiện phép biến đổi Gauss tương tự, thay hàng 2 = hàng 3 + 0.5 \* hàng 2, ta được ma trận sơ cấp tương ứng

* Lấy ma trận trên nhân với A

* Tương tự, nếu ta thực hiện 1 lần phép biến đổi Gauss nào đó trên 1 cột nào đó của A, thì kết quả tương đương với việc lấy A nhân với 1 ma trận sơ cấp n x n ứng với phép biến đổi Gauss đó
* Ví dụ

* Sử dụng phép biến đổi Gauss thay cột 1 = cột 3 + cột 2 + cột 1, ta được

* Cũng lấy ma trận I 3 x 3, và thực hiện phép biến đổi Gauss tương tự, thay cột 1 = cột 3 + cột 2 + cột 1, ta được ma trận sơ cấp tương ứng

* Lấy A nhân ma trận trên

1. Ma Trận Hermitian (Hermitian Matrix, Self Adjoint Matrix)?

* Là ma trận vuông A bằng chính chuyển vị liên hợp phức của nó

* Cho ma trận B bất kì, BBT luôn là 1 ma trận bằng chính chuyển vị của nó, còn BB\* luôn là ma trận Hermitian
* Nếu B vuông, B + BT luôn là 1 ma trận bằng chính chuyển vị của nó, còn B + B\* luôn là ma trận Hermitian
* Eigen Value của ma trận Hermitian luôn là số thực
* Chứng minh
* Gọi Eigen Value là λ, Eigen Vector là v, ta luôn có

* Như vậy λ phải là số thực
* Các cặp Eigen Vector độc lập tuyến tính với Eigen Value khác nhau của ma trận Hermitian luôn vuông góc với nhau
* Chứng minh
* Cho x và y là 2 Eigen Vector độc lập tuyến tính bất kì của A, a và b là Eigen Value tương ứng của x và y, a khác b, ta có

* Như vậy rõ ràng x phải vuông góc y

1. Ma Trận Đối Xứng Thực (Real Symmetric Matrix)?

* Là ma trận Hermitian A với phần tử thực, luôn khả chéo hóa trực giao
* Các Eigen Value luôn là số thực
* Các Eigen Vector độc lập tuyến tính vuông góc với nhau

1. Ma Trận Trực Giao (Orthogonal Matrix)?

* Là ma trận vuông A n x n thỏa mãn

* I là ma trận Identity
* Định nghĩa trên tương đương việc chuyển vị của A cũng chính là nghịch đảo của A
* Đồng thời, nó cũng tương đương việc các cột của A đều là Vector có độ dài = 1 và vuông góc với nhau từng đôi một, và các hàng của A cũng đều là Vector có độ dài = 1 và vuông góc với nhau từng đôi một
* Chứng minh điều trên, ta có

* Dễ thấy độ dài mỗi Vector cột trong A phải = 1 và chúng phải vuông góc với nhau
* Mặt khác, ta có

* Như vậy, dễ thấy các Vector hàng trong A cũng phải có độ dài = 1 và chúng phải vuông góc với nhau
* Ma trận trực giao thực chất là ma trận quay nhưng có Flip

1. Ma Trận Xác Định (Definite Matrix)

* Một ma trận A xác định thì chắc chắn nó là ma trận Hermitian, kích thước n x n, tuy nhiên không phải ma trận Hermitian nào cũng xác định
* Được gọi là bán xác định dương (Positive Semi Definite), nếu thỏa mãn

* Được gọi là toàn phương xác định dương (Positive Definite), nếu thỏa mãn

* Một ma trận toàn phương xác định dương thì cũng là bán xác định dương
* Được gọi là bán xác định âm (Negative Semi Definite), nếu thỏa mãn

* Được gọi là toàn phương xác định âm (Negative Definite), nếu thỏa mãn

* Một ma trận toàn phương xác định âm thì cũng là bán xác định âm
* Dễ thấy ma trận không, vừa là bán xác định dương, vừa là bán xác định âm
* Hàm f(x) = x\*Ax được gọi là 1 dạng bậc 2 (Quadratic Form), là 1 ánh xạ từ tọa độ x tới 1 số thực
* Ta có các định lý sau
* A là ma trận toàn phương xác định dương khi và chỉ khi nó là ma trận Hermitian và mọi Eigen Value của nó đều là số thực và > 0
* A là ma trận bán xác định dương khi và chỉ khi nó là ma trận Hermitian và mọi Eigen Value của nó đều là số thực không âm
* A là ma trận toàn phương xác định âm khi và chỉ khi nó là ma trận Hermitian và mọi Eigen Value của nó đều là số thực và < 0
* A là ma trận bán xác định âm khi và chỉ khi nó là ma trận Hermitian và mọi Eigen Value của nó đều là số thực không dương
* A là ma trận Hermitian không xác định (Indefinite) khi và chỉ khi nó có 1 số Eigen Value âm và 1 số Eigen Value dương
* Chứng minh
* Giả sử A là 1 ma trận bán xác định dương, v là 1 Eigen Vector bất kì có Eigen Value là λ, ta có

* Mặt khác, nếu A là ma trận Hermitian n x n, và tất cả Eigen Value của nó không âm, gọi Eigen Value nhỏ nhất là λmin, theo tính chất của thương số Rayleigh, ta có

* Vậy ta đã chứng minh được cả 2 chiều
* Cho ma trận vuông A n x n bất kì, ta luôn có AA\* là bán xác định dương
* Chứng minh

* Vậy AA\* là bán xác định dương

1. Ma Trận Chuẩn (Normal Matrix)?

* Là ma trận vuông A thỏa mãn

* Ma trận Hermitian, ma trận trực giao đều là ma trận chuẩn

1. Ma Trận Hoán Vị (Permutation Matrix)?

* Là ma trận A nhận được khi hoán vị hàng hoặc cột của 1 ma trận Identity
* Dùng để hoán vị tọa độ của 1 điểm
* Ví dụ

* Vì ma trận hoán vị cũng là ma trận trực giao nên ta có

Matrix Form – Dạng Của Ma Trận:

1. Dạng (Form) Của Ma Trậnì?

* Cho ma trận A, dùng các phép biến đổi Gauss để tạo ra ma trận B, khi đó B là 1 dạng của A, nói cách hàng A và B tương đương hàng (Row Equivalent)

1. Tương Đương Hàng (Row Equivalent)?

* 2 ma trận gọi là tương đương hàng nếu có thể dùng các phép biến đổi Gauss để chuyển đổi qua lại giữa 2 ma trận
* Tập hợp tất cả các ma trận tương đương hàng tạo thành 1 lớp

1. Dạng Hàng Bậc Thang (Row Echelon Form) Và Dạng Hàng Bậc Thang Rút Gọn (Row Reduced Echelon Form, Row Canonical Form)?

* Xét dạng hàng bậc thang
* Từ trên xuống, xét mỗi hàng, phần tử đầu tiên của hàng khác 0 gọi là phần tử cơ sở, vị trí của số này gọi là vị trí cơ sở (Pivot Position), biến ứng với nó gọi là biến cơ sở (Leading Variable), và cột chứa nó là cột cơ sở (Pivot Column)
* Cột sơ sở ở hàng tiếp theo phải ở bên phải cột cơ sở của hàng hiện tại
* Các biến không phải là biến cơ sở của bất kì dòng nào gọi là biến tự do

(Free Variable)

* Nếu tất cả cột cơ sở đều là One Hot Vector thì là dạng hàng bậc thang rút gọn
* Mỗi ma trận sẽ có nhiều dạng hàng bậc thang nhưng chỉ có duy nhất 1 dạng hàng bậc thang rút gọn
* Ma trận hàng bậc thang rút gọn còn được gọi là ma trận chính tắc của tất cả ma trận tương đương hàng với nó
* Ví dụ

* Dạng hàng bậc thang rút gọn của M là

* Để ý thấy, nếu sắp xếp lại các cột của dạng hàng bậc thang rút gọn, ta luôn có dạng

* I là ma trận Identity
* F là ma trận có cùng số hàng với I
* 0 là ma trận 0
* Từ công thức trên, ta dễ dàng suy ra được ma trận Null Space, nói cách khác là ma trận mà các cột của nó đều là các nghiệm đặc biệt của hệ thuần nhất ứng với ma trận đang xét đã được sắp xếp cột

* I’ cũng là ma trận Identity, khác I
* Ví dụ

Complex Matrix – Ma Trận Phức:

1. Ma Trận Liên Hợp Phức (Conjugate Matrix) Của 1 Ma Trận Nào Đó?

* Cho ma trận A bất kì, lấy liên hợp các phần tử phức của A được ma trận liên hợp phức của A
* Ví dụ

1. Chuyển Vị Liên Hợp Phức (Conjugate Transpose, Hermitian Transpose) Của 1 Ma Trận Nào Đó?

* Cho ma trận A bất kì, lấy liên hợp các phần tử phức của A rồi chuyển vị được chuyển vị liên hợp phức của A
* Ví dụ

1. Thương Số Rayleigh?

* Cho ma trận Hermitian A n x n và Vector x, khi đó, thương số Rayleigh là

* Bản chất của thương số Rayleigh chính là trung bình có trọng số của tất cả Eigen Value
* Chứng minh
* Vì A là ma trận Hermitian nên có các Eigen Vector độc lập tuyến tính vuông góc với nhau từng đôi một, chọn tất cả các Eigen Vector độc lập tuyến tính với độ dài = 1 làm thành cột của ma trận chuyển đổi Mod của A, gọi là Q, rõ ràng Q là 1 ma trận trực giao, ta có

* Λ là ma trận đường chéo vuông chứa λ là Eigen Value của A

* Rõ ràng đây chính là trung bình có trọng số của tất cả Eigen Value, trọng số là bình phương tọa độ x ứng với Basis là Q
* Dễ thấy nếu x là Eigen Vector thì thương số Rayleigh sẽ là Eigen Value tương ứng
* Đồng thời, do trọng số là bình phương nên chắc chắn dương, do đó Min và Max của thương số Rayleigh lần lượt là Eigen Value nhỏ nhất và Eigen Value lớn nhất

1. Tích Vô Hướng Vector Phức?

* Cho 2 Vector phức x và y, ta có tích vô hướng của x với y là

* Dễ thấy tích vô hướng của Vector phức không có tính chất giao hoán
* x và y vuông góc chỉ khi

Vector Space – Không Gian Vector:

1. Không Gian Vector Là Gì?

* 1 không gian Vector trên 1 trường F là 1 tập hợp không rỗng V kết hợp với 2 toán tử là phép cộng giữa 2 phần tử trong V, kí hiệu là “+” và phép nhân giữa 1 phần tử trong F với 1 phần tử trong V để cho ra 1 phần tử trong V, kí hiệu là av, a thuộc F, v thuộc V, đồng thời thỏa mãn các tính chất sau, giả sử Identity của F ứng với toán tử cộng là e1, ứng với toán tử nhân là e2, của (V, +) là e3, phần tử đối xứng với e2 trong phép cộng kí hiệu là –e2, phần tử đối xứng với phần tử v nào đó trong V kí hiệu là –v

(V, +) phải là nhóm giao hoán

* Hệ quả

* Phần tử trong V gọi là Vector

1. Không Gian Tích Trong (Inner Product Space)?

* Là 1 không gian Vector V trên trường F là trường số thực R hoặc số phức C kèm theo 1 toán tử tích trong, toán tử này áp dụng lên 2 Vector trong V để cho ra 1 phần tử trong F, kí hiệu là 〈u,v〉, u và v là 2 Vector trong V, đồng thời với mọi Vector u, v, w trong V và với mọi phần tử a, b trong F, chúng phải thỏa mãn toàn bộ các tính chất sau, lưu ý 1 dấu gạch trên đầu là liên hợp phức, 2 dấu gạch trên đầu là lấy phần thực, Identity của của (V, +) là e

* Hệ quả

* Chiều dài (Norm) của Vector v trong gian tích trong V được định nghĩa bằng biểu thức

* Khoảng cách giữa 2 Vector u và v trong không gian tích trong V được định nghĩa bằng biểu thức sau, phần tử đối xứng v trong V kí hiệu là –v

* 2 Vector u và v trong không gian tích trong V được gọi là vuông góc khi tích trong của chúng = 0
* Vector u vuông góc với tập hợp Vector A trong V chỉ khi u vuông góc với từng phần tử của A
* Tập hợp Vector M gọi là 1 họ trực giao của V khi mỗi cặp Vector trong M đều vuông góc với nhau
* M gọi là họ trực chuẩn khi tất cả Vector trong M đều có chiều dài = 1
* Tích vô hướng như ta đã biết ở cấp 2 gọi là tích trong chính tắc hay tích vô hướng chính tắc

1. Không Gian Vector Euclidean (Euclidean Vector Space)?

* Là không gian tích trong trên trường số thực R với số chiều hữu hạn
* Góc giữa 2 Vector u và v trong 1 không gian Vector Euclidean được định nghĩa bằng biểu thức sau

1. Tổ Hợp Tuyến Tính (Linear Combination)?

* Cho không gian Vector V trên trường F, cho A = [v1, v2, …, vn] là 1 dãy các Vector nào đó trong V, các Vector có thể trùng nhau, cho B = [a1, a2, …, an] là 1 dãy các phần tử nào đó trong F, các phần tử có thể trùng nhau, khi này tổ hợp tuyến tính v của A với B là

* Nếu tất cả phần tử trong B đều = Identity của F ứng với phép cộng, thì tổ hợp tuyến tính trên gọi là tổ hợp tuyến tính tầm thường (Trivial Linear Combination)
* Còn nếu chỉ cần chứa 1 phần tử khác Identity, thì gọi là tổ hợp tuyến tính không tầm thường (Non Trivial Linear Combination)
* Nếu tồn tại B sao cho tổ hợp trên là không tầm thường và cho ra v = Identity của (V, +) thì ta nói A phụ thuộc tuyến tính (Linear Dependent)
* Ngược lại, ta nói A độc lập tuyến tính (Linear Independent)
* Nếu A là 1 tập hợp, tức là chứa các phần tử khác nhau, thì tất cả tổ hợp tuyến của A với B bất kì gọi là họ Vector A hay Span của A, kí hiệu là Span(A) hoặc L(A), nếu A rỗng thì Span(A) chỉ chứa phần tử Identity của (V, +), kí hiệu {0}
* Hạng của Span(A) = số tối đại các Vector độc lập tuyến tính của A, nghĩa là số chiều không gian nó Span
* A được gọi là tập sinh (Spanning Set) của V nếu mọi phần tử trong V = tổ hợp tuyến tính nào đó của A

1. Không Gian Con (Subspace)?

* Là tập con của 1 không gian Vector V nào đó và đồng thời cũng là 1 không gian Vector
* Không gian con này được gọi là bình thường (Proper) nếu nó không phải là V và được gọi là tầm thường (Trivial) nếu nó = {0}
* Cho 2 không gian con nào đó là A và B
* Giao của 2 không gian con A ∩ B = không gian con khác bé hơn hoặc bằng 2 không gian con đầu
* Lấy tổng 2 phần tử bất kì tương ứng trong 2 không gian con, ta sẽ Span 1 không gian con khác lớn hơn hoặc bằng 2 không gian con đầu = A + B
* Số chiều Span bởi A + B = số chiều Span bởi A + số chiều Span bởi B – số chiều Span bởi A ∩ B
* Tập hợp tất cả Vector vuông góc với không gian con A trong V gọi là bù vuông góc của A trong V, ví dụ đường thẳng vuông góc mặt phẳng

* Số chiều của A + số chiều của bù vuông góc của A = số chiều của V
* Bù vuông góc cũng là 1 không gian con trong V
* Cho 1 Vector v bất kì trong V, cho không gian con A và bù vuông góc của A là B, khi này tồn tại duy nhất cặp Vector f, g, f thuộc A, g thuộc B, sao cho f + g = v, khi này f được gọi là hình chiếu vuông góc của v xuống A, kí hiệu f = prAv , và g được gọi là hình chiếu vuông góc của v xuống B, kí hiệu prBv
* Khoảng cách từ giữa v và A = chiều dài của g, tương tự, khoảng cách giữa v và B = chiều dài của f
* Ví dụ, A là 1 đường thẳng qua gốc tọa độ trong không gian 3D, B là mặt phẳng vuông góc A, cũng qua gốc tọa độ, cho 1 điểm M đâu đó, khi này dễ thấy M = Vector hình chiếu của M lên A + Vector hình chiếu của M lên B

1. Cơ Sở (Basis)?

* B được gọi là 1 cơ sở của không gian Vector V nếu nó là tập sinh của V và độc lập tuyến tính
* 1 không gian Vector có thể có nhiều Basis, và tất cả các Basis này đều có cùng số phần tử, gọi là số chiều của không gian Vector (Dimension) = dim(V)
* Những cơ sở có cùng phần tử nhưng khác thứ tự thì khác nhau
* B gọi là hữu hạn nếu số chiều nó Span là hữu hạn

1. Cơ Sở Chính Tắc (Standard Basis)?

* Còn gọi là cơ sở sắp thứ tự, là cơ sở chứa toàn One Hot Vector
* Ví dụ
* Cơ sở chính tắc của không gian tọa độ thực

1. Biểu Diễn Của 1 Vector Ứng Với 1 Cơ Sở?

* Cho Vector v, cơ sở B, biểu diễn của v ứng với B là 1 Vector mà các phần tử của nó là tọa độ của v ứng với B
* Kí hiệu

1. Cách Chuyển Hệ Tọa Độ?

* Cho Vector v trong không gian tọa độ thực, v đang có tọa độ ứng với cơ sở chính tắc, ta muốn biểu diễn tọa độ của v theo cơ sở khác thì dùng công thức

* B là ma trận có các cột là cơ sở ta muốn biểu diễn tọa độ theo, gọi là ma trận chuyển cơ sở từ cơ sở chính tắc tới cơ sở B
* Ví dụ

1. Cách Tạo Cơ Sở Vuông Góc Bằng Quá Trình Gram Schmidt?

* Ban đầu ta có hệ Vector v1, v2, …, vn, và ta muốn tạo ra hệ Vector khác sao cho mỗi Vector trong đó vuông góc với nhau từng đôi một, hệ này là u1, u2, …, un
* Đầu tiên u1 = v1, tiếp theo, u2 sẽ là cái Vector đồng phẳng với u1 và v2, đồng thời vuông góc u1, u2 đạt được = cách trừ v2 đi 1 lượng u1 sao cho nó vừa đủ vuông góc với u1

* Tiếp tục u3 sẽ đạt được = cách trừ v3 đi 1 lượng u2 và u1 sao cho nó vừa đủ vuông góc với cả u2 và u1

* Tổng quát hóa, un sẽ được tính như sau

* Hệ Vector u tạo thành 1 hệ vuông góc, đồng thời có Span = Span của hệ Vector cũ v, trong quá trình này, nếu un tính được = 0, thì un phụ thuộc tuyến tính vào những Vector trước đó, nên quá trình Gram Schmidt còn được dùng để Check xem hệ Vector có độc lập tuyến tính không
* Sau khi có được hệ Vector u, ta sẽ chuẩn hóa nó để độ dài mỗi Vector = 1, gọi ma trận Q có các cột là Vector u đã được chuẩn hóa, khi đó, do Span của Q = Span của hệ Vector v, nên ma trận chiếu vuông góc vào Q cũng chính là ma trận chiếu vuông góc vào Span của hệ Vector v, dùng công thức ma trận chiếu vuông góc, ta có ma trận chiếu là

* Như vậy, quá trình Gram Schmidt còn được dùng để thiết lập ma trận chiếu vuông góc vào hệ Vector ban đầu

1. Một Số Không Gian Vector?

* Không gian Vector tọa độ thực n chiều, mọi không gian Vector đều có thể quy về không gian này bằng cách chuyển cơ sở

* Không gian Vector với Vector là các đa thức bậc n theo biến x trên trường số thực

* Không gian Vector với Vector là các ma trận m x n với m khác n trên trường F

* Ví dụ các ma trận có kích thước 3 x 2 trên trường số thực

* Trường hợp ma trận vuông cấp n kí hiệu như sau

Product – Tích:

1. Tích Hadamard (Hadamard Product)?

* Là phép nhân từng phần tử của ma trận này với phần tử tương ứng của ma trận kia
* Ví dụ

1. Tích Kronecker (Kronecker Product)?

* Tích Kronecker giữa Tensor A và Tensor B là phép thay thế mỗi phần tử C của A bằng 1 Tensor có giá trị = CB
* Ví dụ

1. Tích Ngoài (Outer Product) Của 2 Tensor?

* Giả sử Tensor A có Shape là (7, 8, 9), Tensor B có Shape là (10, 11, 12, 13, 15), gọi C là tích ngoài của A với B, ta có

* a, b, c, e, f, g, h, i là các Index
* Ví dụ

1. Tensor Nén (Tensor Contraction)?

* Ví dụ Tensor A có Shape là (7, 8, 9), Tensor B có Shape là (10, 11, 12, 8, 15), gọi C là Tensor Nén của A với B, chiều nén tương ứng là chiều thứ 2 của A và thứ 4 của B, ta có

* a, c, e, f, g, i là các Index

1. Tích Nêm (Wedge Product) Bản Chất Là Gì?

* Tích nêm của 2 Vector u và v sẽ tạo 1 Vector ở chiều không gian khác gọi là không gian ngoài bình phương (Exterior Square) mà khi lấy Module của nó sẽ cho ra diện tích hình bình hành tạo bởi 2 Vector ấy, kí hiệu u ∧ v
* Vì diện tích của hình bình hành tạo bởi 2 Vector trùng nhau = 0 nên

* Hơn nữa, kéo dài 1 Vector bao nhiêu lần thì diện tích cũng tăng bấy nhiêu lần

* Tưởng tượng u, v, w tạo thành hình hộp chữ nhật để chứng minh tính chất phân phối của tích nêm với trường hợp hình hộp chữ nhật, rồi dùng ma trận biến đổi thành hình hộp bình hành bất kì để áp dụng cho u, v, w bất kì, ta được tính chất phân phối tổng quát

* Từ 2 tính chất trên, ta có

* Tính chất kết hợp của tích nêm

* Tích nêm của n Vector trong không gian m chiều sẽ có số phần tử là tổ hợp chập n của m, và Basis có dạng tích nêm của n Vector đơn vị

1. Tích Có Hướng Có Phải Tích Nêm?

* Không, mặc dù giá trị các phần tử giống nhau, nhưng tích có hướng tạo Vector có Basis là i, j, k, …, còn tích nêm tạo Vector có Basis là i ∧ j, i ∧ k, j ∧ k, …

1. Ví Dụ Tính Tích Nêm?

* Tìm diện tích hình bình hành tạo ra từ 2 Vector trong không gian 4 chiều sau, gọi các Vector đơn vị của không gian này là i, j, k, l

* Vector trong chiều không gian ngoài bình phương có Module là diện tích của hình bình hành này là

* Diện tích của hình bình hành này là

1. Mẹo Tính Tích Nêm?

* Giả sử ta muốn tính A = x1 ∧ x2 ∧ x3 ∧ … ∧ xn, đây đều là những Vector trong không gian m chiều, n không vượt quá m, xếp các Vector này thành ma trận sau

* Để tính hệ số của 1 Basis bất kì của A, ta xác định Basis đó là tích nêm của những Vector đơn vị nào, lưu ý các Vector đơn vị này sắp xếp theo thứ tự tăng dần, sau đó giữ lại các cột tương ứng với những Vector đơn vị này, loại bỏ các cột còn lại, rồi tính định thức của ma trận nhận được, ta được hệ số

Gaussian Elimination – Phép Khử Gauss:

1. Phép Khử Gauss Trên 1 Ma Trận?

* Là 1 trong 2 thao tác sau
* Thế 1 hàng/cột = tổ hợp tuyến tính của tất cả các hàng/cột, trong đó hệ số của hàng/cột được thế phải khác 0
* Tráo các hàng/cột

1. Phép Khử Gauss Làm Biến Đổi Và Không Làm Biến Đổi Những Tính Chất Nào Của Ma Trận?

* Xét trường hợp chỉ khử hàng
* Khi dùng phép khử Gauss để đạt được mục đích là đạt được dạng hàng bậc thang rút gọn, ta sẽ vô tình từ từ loại bỏ mọi quan hệ tuyến tính giữa các hàng, nghĩa là dạng ma trận cuối cùng nhận được, bất kể hàng nào của nó cũng không thể là tổ hợp tuyến tính của các hàng còn lại
* Trong suốt quá trình khử, không gian hàng và Null Space của ma trận không thay đổi, nhưng không gian cột và Left Null Space có thể thay đổi
* Đồng thời, các cột trong ma trận ban đầu có vị trí là vị trí các cột cơ sở ở dạng hàng bậc thang rút gọn cuối cùng sẽ tạo thành Basis của không gian cột của ma trận ban đầu
* Xét trường hợp chỉ khử cột
* Y chang khử hàng, đổi chữ “hàng” thành “cột”, “Null Space” thành “Left Null Space” và ngược lại
* Nếu sử dụng cả khử dòng và khử cột cùng lúc thì có thể làm biến đổi mọi tính chất của ma trận

1. Dùng Phép Khử Gauss Để Tìm Nghịch Đảo Của Ma Trận?

* Giả sử A là ma trận vuông khả nghịch m x m, I là ma trận Identity m x m, để tìm nghịch đảo của A, ta thực hiện phép khử Gauss theo hàng trên ma trận mở rộng (A | I) để được dạng hàng bậc thang rút gọn (I | A-1), khi đó phần ma trận m x m bên phải cùng của R chính là nghịch đảo của A
* Ví dụ

* Chứng minh tại sao cách này lại có thể tìm được nghịch đảo
* Khi dùng phép khử Gauss lên ma trận mở rộng (A | I) kiểu gì ta cũng được dạng hàng bậc thang rút gọn (I | B), B là ma trận nào đó
* Vì khi dùng phép khử Gauss theo hàng, Null Space của (A | I) không thay đổi, hay ma trận Null Space của (A | I) y chang của (I | B), dễ thấy ma trận này là

* Như đã nói ở trên, vì đây cũng là ma trận Null Space của (A | I), nên

* Vậy B chính là nghịch đảo của A
* Trong toàn bộ quá trình biến đổi để tìm nghịch đảo
* Số phép nhân, chia được thực hiện = m3
* Số phép cộng, trừ được thực hiện = m3 – 2m2 + m

1. Dùng Phép Khử Gauss Để Tính Định Thức?

* Cho ma trận vuông A, cho a là 1 số thực bất kì, lấy 1 hàng trong A + a lần 1 hàng khác, định thức của A sẽ không thay đổi, đồng thời tất cả Eigen Value của A cũng không đổi dấu, giống như hình hộp bình hành tạo ra từ 3 Vector, bạn tịnh tiến đầu của 1 Vector nào đó theo 1 Vector khác thì rõ ràng thể tích của hình hộp bình hành không đổi
* Sử dụng phép biến đổi trên, chỉ cần ta đưa A về dạng ma trận tam giác trên, rồi lấy tích tất cả phần tử trên đường chéo chính của ma trận đó thì được định thức của A, và dấu của các phần tử này cũng chính là dấu của các Eigen Value của A
* Nếu nhân 1 hàng hoặc cột với k thì định thức sẽ tăng k lần
* Nếu hoán vị 2 hàng/cột bất kì, định thức đổi dấu

1. Trình Bày Phép Khử Gauss Theo Hàng Để Giải Hệ Phương Trình?

* Ví dụ
* Nhân hàng 1 với 4

* Cộng –1 lần hàng 1 vào hàng 2 và –2 lần hàng 1 vào hàng 3

* Tráo hàng 2 với hàng 3

* Giải từ dưới lên

1. Biểu Diễn Nghiệm Của Hệ Phương Trình Vô Số Nghiệm?

* Ví dụ

1. Giải Hệ Phương Trình Dưới Dạng Ma Trận?

* Dùng ma trận mở rộng rồi thực hiện phép khử Gauss theo hàng để đưa ma trận về dạng hàng bậc thang
* Ví dụ
* Hệ phương trình ban đầu

* Biến đổi hệ phương trình dưới dạng ma trận mở rộng

1. Biểu Diễn Tập Nghiệm Vô Hạn Dưới Dạng Ma Trận Và Vector?

* Ví dụ

* Dạng ma trận của tập nghiệm

* Dạng Vector của tập nghiệm

* Dễ thấy tập nghiệm này Span 1 mặt phẳng trong không gian 4 chiều

1. Hệ Thuần Nhất?

* Là hệ phương trình mà tất cả hệ số tự do trong tất cả phương trình đều = 0
* Ví dụ

* Xét hệ thuần nhất dạng tổng quát

* A là 1 ma trận hình chữ nhật
* x là 1 Vector dọc, là ẩn cần tìm
* Giả sử có vô số nghiệm, khi đó tập hợp tất cả các nghiệm thỏa mãn phương trình trên sao cho chỉ có duy nhất 1 biến tự do có giá trị = 1, còn các biến tự do còn lại có giá trị = 0, gọi là các nghiệm đặc biệt (Special Solution)
* Tổ hợp tuyến tính của tất cả các nghiệm đặc biệt tạo thành Null Space hay tập nghiệm của phương trình này
* Hệ thuần nhất luôn có nghiệm = tất cả ẩn = 0, nghiệm này gọi là nghiệm tầm thường (Trivial Solution)

1. Mặt Phẳng Đa Chiều?

* Trong không gian n chiều, cho k Vector không đồng phẳng, k <= n, chọn 1 điểm cố định rồi từ điểm này Span k Vector kia, được mặt phẳng k chiều
* Ví dụ
* Đường thẳng là mặt phẳng 1 chiều

* Mặt phẳng 2 chiều trong không gian 4 chiều

1. Giải Pháp Bình Phương Tối Thiểu (Least Squares Solution) Cho Phương Trình Tuyến Tính Vô Nghiệm?

* Cho A là 1 ma trận chữ nhật bất kì, B là 1 Vector nào đó, tìm Vector X để

* Giả sử phương trình này vô nghiệm, vậy thì làm sao ta tìm ra X để AX gần với B nhất, để ý thấy, khi X chịu tác dụng của A, nó sẽ bị ánh xạ tới 1 điểm đâu đó trong không gian cột của A, gọi điểm này là X’, như vậy bài toán đưa về tìm X để khoảng cách X’B ngắn nhất, vâng X’ chính là hình chiếu của B lên không gian cột của A, kết hợp công thức ma trận chiếu vuông góc P, ta được phương trình

* A-1 là chính là nghịch đảo bên trái của A
* Phương pháp này áp dụng cho cả trường hợp B và X là ma trận

1. Hệ Không Tương Thích (Inconsistent System) Và Hệ Tương Thích (Consistent System)?

* Hệ không tương thích là hệ phương trình vô nghiệm
* Hệ tương thích thì có 1 nghiệm hoặc vô số nghiệm

1. Xác Định Nhanh Số Nghiệm Của Hệ Phương Trình Sử Dụng Định Lý Kronecker Capelli (Rouche Capelli)?

* Cho A là ma trận hệ số của hệ phương trình, B là ma trận cột hệ số tự do, A phải là ma trận vuông n x n, với n là số ẩn
* Nếu hạng của ma trận bổ sung là r(A|B) không bằng hạng của A là r(A), tức là r(A|B) > r(A), thì B đang nằm ở chiều không gian khác so với không gian cột của A, do đó sẽ không có cách nào để A ánh xạ 1 điểm tới B, do đó hệ phương trình vô nghiệm
* Nếu r(A|B) = r(A) = n, thì hệ có nghiệm duy nhất
* Nếu r(A|B) = r(A) < n thì hệ có vô số nghiệm

1. Quy Tắc Cramer?

* Cho A là ma trận vuông n x n không suy biến, X và B là 2 ma trận n x m bất kì sao cho thỏa mãn AX = B
* Khi này, gọi C là ma trận con được tạo thành từ các hàng thứ i1, i2, …, ik và các cột thứ j1, j2, …, jk của X
* Ví dụ

* Dễ thấy C được tạo thành từ các hàng 2, 4, 6 và các cột 2, 4, 5 của X
* Gọi D là ma trận nhận được bằng cách lấy A rồi thay các cột thứ i1, i2, …, ik trong nó thành các cột thứ j1, j2, …, jk tương ứng trong B, tiếp tục ở ví dụ trên, ta sẽ thay cột 2, 4, 6 của A bằng các cột 2, 4, 5 của B
* Khi này, ta có đẳng thức

Model – Mô Hình:

1. Chuỗi Markov (Markov Chain)?

* Là 1 mô hình mô tả 1 chuỗi các sự kiện, mà tỉ lệ sự kiện tiếp theo xảy ra chỉ phụ thuộc vào trạng thái hiện tại, số trạng thái là hữu hạn
* Ví dụ



* Như trong hình, giả sử ta đang đứng ở clouds, thì đéo cần biết quá khứ ta đã làm gì, ta luôn có 30% tỉ lệ chuyển thành sun, 30% chuyển thành rain và 40% đéo biến đổi, hay khi đứng tại sun, đéo cần biết trước đó ta đứng ở đâu, ta luôn có 10% tỉ lệ chuyển thành rain ở bước tiếp theo, 50% tỉ lệ chuyển thành clouds và 40% tỉ lệ đéo biến đổi
* Mỗi bước chuyển trạng thái sẽ cách nhau 1 khoảng thời gian cố định

1. Ma Trận Ngẫu Nhiên (Stochastic Matrix)?

* Còn gọi là ma trận Markov
* Giả sử ta gieo vào vị trí các trạng thái trong chuỗi Markov một số lượng vật thể nhất định, thì bài toán đặt ra là khi thời gian chạy đến 1 lúc nào đó, số lượng vật thể ứng với mỗi trạng thái là bao nhiêu
* Cho X là Vector, mỗi phần tử của nó tương ứng là số lượng vật thể của trạng thái 1, 2, 3, …, lúc mới gieo vào, M là ma trận Markov ứng với chuỗi Markov này, Si là trạng thái thứ i
* Hàng đầu tiên của M có các giá trị lần lượt là tỉ lệ số vật thể ở S1 giữ nguyên trạng thái, tỉ lệ số vật thể ở S2 chuyển sang S1, tỉ lệ số vật thể ở S3 chuyển sang S1, …
* Hàng thứ 2 tương tự có các giá trị lần lượt là tỉ lệ số vật thể ở S1 chuyển sang S2, tỉ lệ số vật thể ở S2 giữ nguyên trạng thái, tỉ lệ số vật thể ở S3 chuyển sang S2, …
* …
* Cứ như vậy, 1 cột của M sẽ có tổng luôn = 1, do chứa các giá trị là tỉ lệ vật thể từ S1 tới toàn bộ các trạng thái
* Sau n bước, số vật thể ở mỗi trạng thái là

* Khi n dần tới vô tận, thì Mn dần hội tụ, và ta được trạng thái cân bằng

1. Mô Hình Leslie?

* Là mô hình cho cơ cấu dân số của 1 quần thể theo độ tuổi
* Giả sử quần thể được chia làm các lớp tuổi bằng nhau lần lượt là lớp tuổi 1, lớp tuổi 2, …, ví dụ từ 0 đến 5 tuổi là lớp tuổi 1, từ 5 đến 10 tuổi là lớp tuổi 2, …
* Gọi si và fi lần lượt là tỉ lệ sống sót và tỉ lệ sinh của các cá thể thuộc lớp tuổi i, khi sinh con thì con sẽ thuộc lớp tuổi 1, giả sử có n lớp tuổi, khoảng cách giữa các lớp là k năm, lớp tuổi cuối sau k nữa năm sẽ chết
* Ma trận Leslie ứng với quần thể này sẽ có kích thước n x n, hàng đầu tiên sẽ chứa các phần tử lần lượt là f1, f2, …, fn, khu vực (n – 1) x (n – 1) ở dưới hàng 1, tay trái, có dạng 1 ma trận đường chéo, với các phần tử trên đường chéo là s1, s2, …, sn – 1, phần còn lại chứa toàn 0, ví dụ

* Cho Vector X chứa các phần tử lần lượt là số cá thể ở lớp tuổi 1, 2, …, n thời điểm hiện tại, thì sau m \* k năm nữa, số cá thể của từng lớp tuổi là

* Nếu lớp tuổi cuối vẫn đéo chết và có tỉ lệ sống tiếp là sn, thì trong L, phần tử cuối cùng sẽ chuyển từ 0 thành sn

1. Mô hình Input Output?

* Giả sử ta có các sản phẩm x1, x2, …, xn, mỗi sản phẩm này có công thức chế tạo như sau

x1 = a11x1 + a21x2 + a31x3 + … + an1xn (1)

x2 = a12x1 + a22x2 + a32x3 + … + an2xn (2)

…

xn = a1nx1 + a2nx2 + a3nx3 + … + annxn (n)

* Trong đó aij = số nguyên liệu xi cần thiết để tạo ra 1 sản phẩm xj
* Giả sử chúng ta muỗn sản xuất ra b1 sản phẩm x1, b2 sản phẩm x2, …, bn sản phẩm xn, khi này số nguyên liệu cần thiết sẽ = b1 \* (1) + b2 \* (2) + … + bn \* (n), ta sẽ sử dụng ma trận sau để tính toán dễ hơn

* Dễ thấy cột 1 của A ứng với công thức cho x1, cột 2 cho x2, …
* Cho Vector B là Vector chỉ số lượng sản phẩm muốn tạo ra

* Khi này Vector chỉ số lượng nguyên liệu cần để sản xuất ra lượng sản phẩm B là

* Nghĩa là để sản xuất ra b1 sản phẩm x1, b2 sản phẩm x2, …, bn sản phẩm xn, thì cần c1 sản phẩm x1, c2 sản phẩm x2, …, cn sản phẩm xn
* Lưu ý ở đây ta phải trừ lượng sản phẩm sản xuất được cho lượng nguyên liệu thì mới được lượng sản phẩm lời, ví dụ dùng 3 cục cứt để tạo ra 7 cục cứt, khi này ta đã tạo ra thêm được 4 cục cứt, đây gọi là nhu cầu cuối cùng, là thứ có thể bán kiếm tiền, thay vì bán hết
* Nhu cầu cuối cùng do đó được tính bằng

* Đảo ngược lại bài toán, cho trước nhu cầu cuối cùng và công thức tạo ra sản phẩm, hỏi tổng số lượng sản phẩm sản xuất được, còn gọi là đầu ra, ta có

Outer Product:

Def:

Là tích 2 Vector dưới dạng Matrix

Tác dụng của Vector nào đó lên hướng này biểu diễn trên hướng kia

Ex:

Singular Value:

Def:

Căn 2 của Eigen Value của Matrix hoặc tùy theo matrix

nào có số chiều nhỏ hơn

Sắp xếp các Singular Value theo chiều giảm dần

Ex:

Left Singular Vector:

Def:

Normalized Eigen Vector của Matrix

Ex:

Right Singular Vector:

Def:

Normalized Eigen Vector của Matrix

Ex:

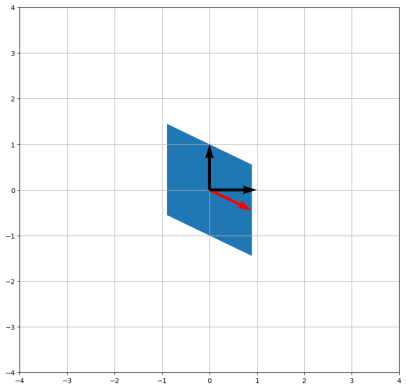
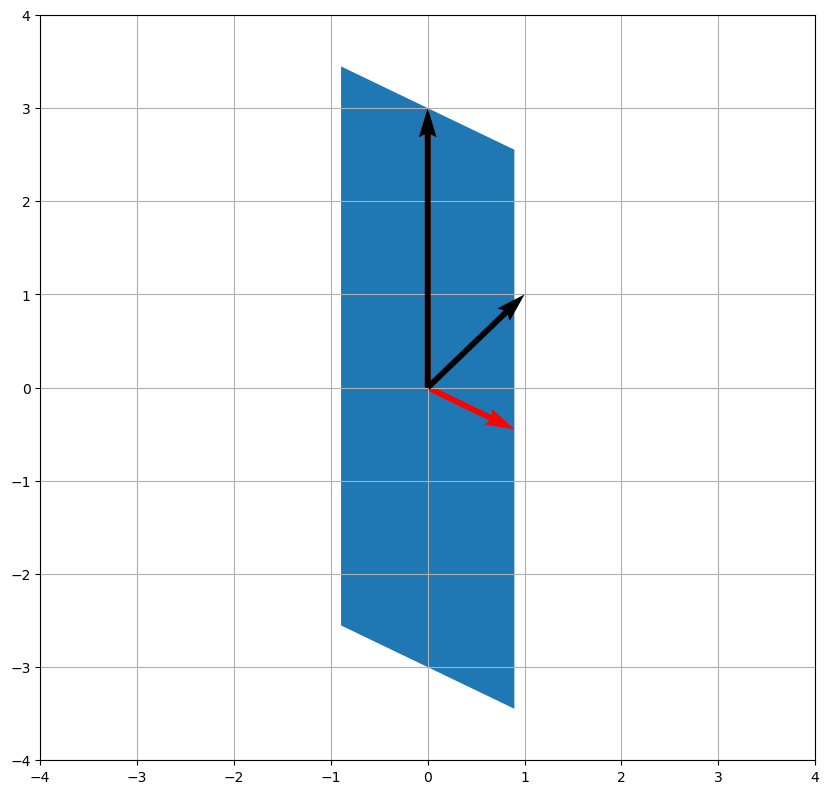
Eigen Value Decomposition:

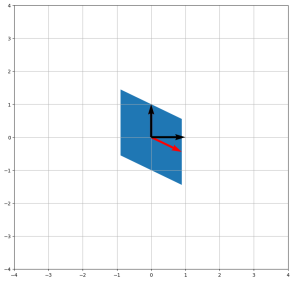
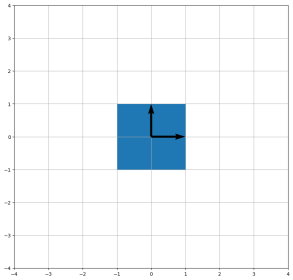
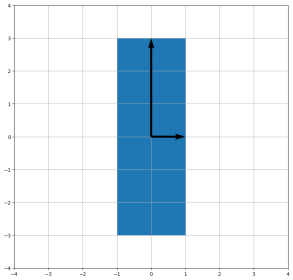
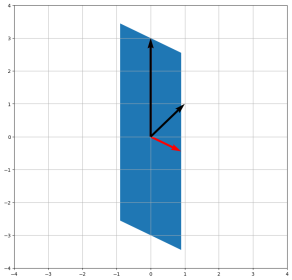
Def:

Phân tích Matrix thành tích của Eigen Vectors Matrix và Eigen

Values Matrix

Ex:





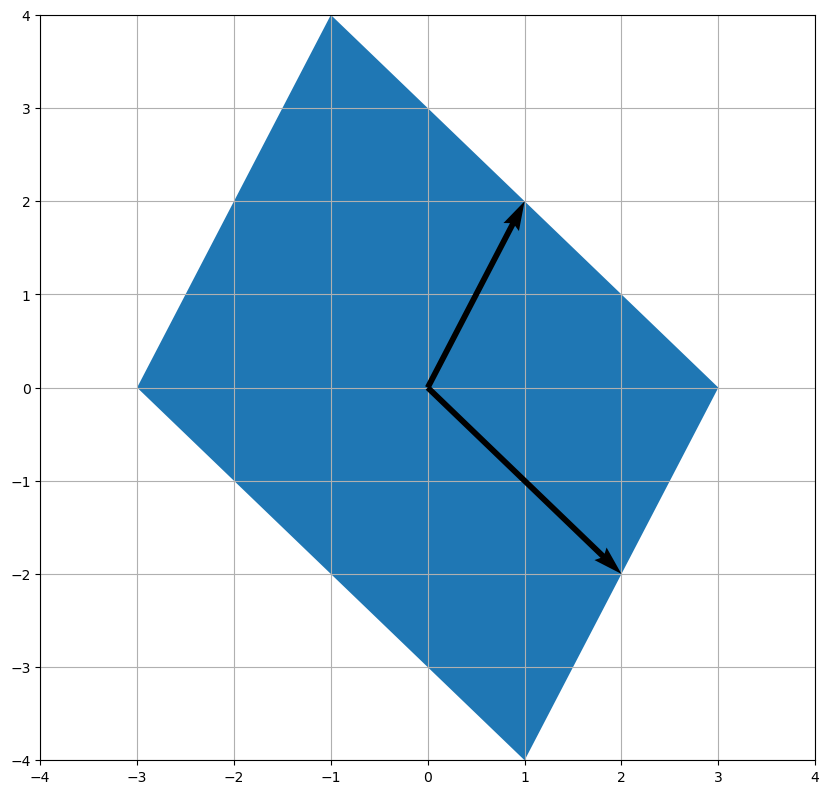
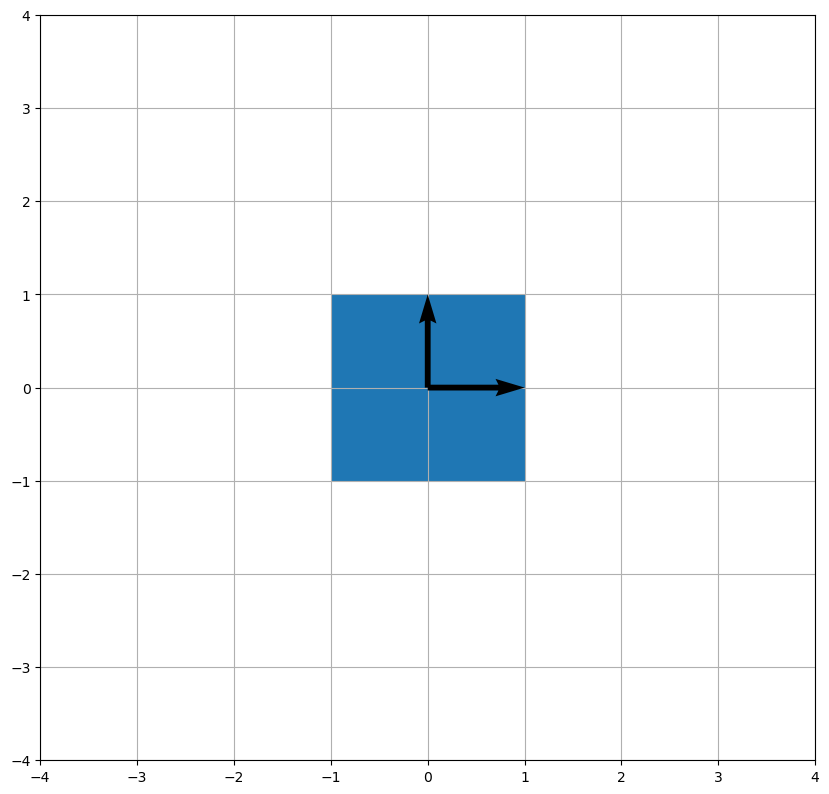
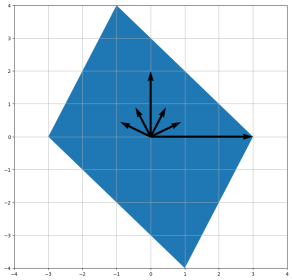
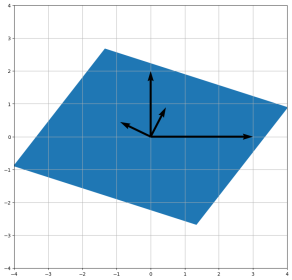
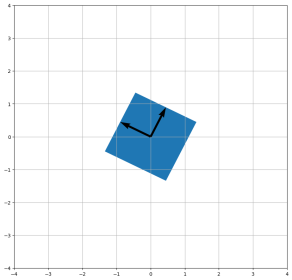
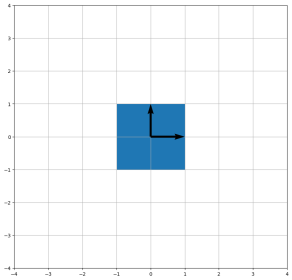
Singular Value Decomposition:

Def:

Phân tích Matrix thành tích của Left Singular Vectors Matrix ,

Singular Values Matrix và Right Singular Vectors Matrix

Ex:



Norm:

Def:

Một hàm Map giá trị bất kì sang giá trị thực không âm như khoảng

cách, chiều dài, độ lớn, …

Two Norm:

Def:

Một Norm tính khoảng cách bằng cách lấy căn 2 của tổng bình

phương các phần tử

Ex:

Low Rank Approximation:

Def:

Mô phỏng gần đúng Matrix bằng một Matrix với Rank nhỏ hơn

Thường sử dụng Singular Value Decomposition để nén ảnh

Phân tích

Vì các Singular Values có giá trị giảm dần nên các Terms càng

về cuối càng không đáng kể nên có thể được loại bỏ

,

Đây là Matrix với Rank k gần với nhất với đạt cực tiểu

Frobenius Norm:

Def:

Một Norm tính khoảng cách bằng cách lấy căn 2 của tổng bình

phương các phần tử trong Matrix , tương đương căn 2 của tổng

bình phương các phần tử trong Singular Values Matrix

Ex:

Linear Independence:

Def:

Matrix mà mỗi Column Vector không thể được tạo ra bằng các

phép toán Linear trên các Column Vector khác

Ex:

không Linear Independent

Rank:

Def:

Số Linear Independent Column hay số chiều không gian sau khi

được tạo ra khi Span các Column Vector trong Matrix

Ex:

Matrix Decomposition: đưa matrix về dạng tích các matrix

Diagonalizable Matrix:matrix M có thể decomposition thành A-1BA, B là diagonal

matrix với entry là eigenvalue, A là matrix có col là eigenvector

Orthogonal Matrix: matrix A mà mỗi row là vector có độ dài 1 và các vector vuông

góc với nhau đôi một

Formula: ATA = I, AT = A-1

Ex: [1/3, 2/3, -2/3]

[-2/3, 2/3, 1/3]

[2/3, 1/3, 2/3]

Lower Triangular Matrix: matrix tam giác mà mọi phần tử bên trên đường chéo = 0

Ex: [1, 0, 0]

[2, 4, 0]

[3, 1, 2]

Positive-definite Matrix: matrix đối xứng A mà mọi vector z <> 0, zTAz > 0

Ex:[1, 0]

[0, 1]

Symmetric Matrix: matrix đối xứng qua diag

Formula:

AT = A

Mọi B, BT + B = Symmetric Matrix

A = UTSU, U là orthogonal matrix, S là diagonal matrix với entry là

Eigenvalue, => eigen vector x(value a), y(value b) orthogonal

ax•y = Ax•y = xTATy = x•Ay = x•by => (a - b)x•y = 0 => x ⊥ y => đpcm

Ex: [1, 2, 4]

[2, 0, 3]

[4, 3, 0]

Sparse Matrix: matrix mà đa số phần tử = 0

Ex: [1, 0, 2]

[0, 0, 3]

[0, 0, 0]

Base Q-Orthogonal: xét vector này có vuông góc với vector khác trong hệ tọa độ

Matrix khác

Formula: uTQv = 0

Ex: A = [2, 3]

[4, 5]

Trong hệ A, vector a [13, -7] vuông góc vector b [2, 1] vì a vuông góc Ab

Jacobian Matrix:

Def:

Ma trận tuyến tính cục bộ của ánh xạ phi tuyến

Formula:

Cho ánh xạ

Khi đó Jacobian Matrix

Ex:

Cho ánh xạ

Khi đó Jacobian Matrix

Ví dụ tại vị trí có tọa độ

Intuation:

