Dimensionality Reduction (giảm chiều dữ liệu), như đã được đề cập một vài lần trong blog, là một trong những kỹ thuật quan trọng trong Machine Learning. Các feature vectors trong các bài toán thực tế có thể có số chiều rất lớn, tới vài nghìn. Ngoài ra, số lượng các điểm dữ liệu cũng thường rất lớn. Nếu thực hiện lưu trữ và tính toán trực tiếp trên dữ liệu có số chiều cao này thì sẽ gặp khó khăn cả về việc lưu trữ và tốc độ tính toán. Vì vậy, giảm số chiều dữ liệu là một bước quan trọng trong nhiều bài toán. Đây cũng được coi là một phương pháp nén dữ liệu.

Dimensionality Reduction, nói một cách đơn giản, là việc đi tìm một hàm số, hàm số này lấy đầu vào là một điểm dữ liệu ban đầu x∈RDx∈RD với DD rất lớn, và tạo ra một điểm dữ liệu mới z∈RKz∈RK có số chiều K<DK<D.

Và như thường lệ, tôi sẽ trình bày một phương pháp đơn giản nhất trong các thuật toán Dimensionality Reduction dựa trên một mô hình tuyến tính. Phương pháp này có tên là *Principal Component Analysis* (PCA), tức *Phân tích thành phần chính*. Phương pháp này dựa trên quan sát rằng dữ liệu thường không phân bố ngẫu nhiên trong không gian mà thường phân bố gần các đường/mặt đặc biệt nào đó. PCA xem xét một trường hợp đặc biệt khi các mặt đặc biệt đó có dạng tuyến tính là các không gian con (subspace).

Bài viết này dành cho nhiều đối tượng độc giả khác nhau:

* Nếu bạn cần ôn tập lại các kiến thức về hệ độc lập tuyến tính, kỳ vọng, phương sai, ma trận hiệp phương sai, bạn có thể đọc Mục 2.
* Nếu bạn muốn hiểu nguồn gốc, ý tưởng đứng sau PCA, tại sao PCA lại được thực hiện như vậy, bạn có thể tìm được ở Mục 3.
* Nếu bạn không muốn đi sâu vào toán mà chỉ cần hiểu các bước thực hiện PCA, bạn có thể tới ngay Mục 4.
* Nếu bạn không muốn hiểu các bước thực hiện mà chỉ muốn biết hàm số thực hiện PCA, một vài ứng dụng của PCA, và có thể thêm các phần mở rộng của PCA, bạn có thể thấy PCA phần 2 có ích, dự tính được ra mắt sau đây một tuần.
* Nếu tôi là các bạn, tôi sẽ đọc hết cả bài.

Trước khi đi vào chi tiết của PCA, chúng ta cùng điểm lại một chút về Đại số tuyến tính và Thống kê.

2. Một chút toán

2.1. Norm 2 của ma trận

Chúng ta vẫn thường nhắc nhiều đến [norm cho vector](https://machinelearningcoban.com/math/#-norms-chuan) nhưng chưa thực sự làm việc nhiều với norm của ma trận (ngoài [Frobenius norm](https://machinelearningcoban.com/math/#chuan-cua-ma-tran)). Trong mục này, chúng ta sẽ làm quen với 1 lớp các norm cho ma trận được định nghĩa dựa trên norm của vector. Lớp các norms này còn được gọi là *Induced Norms*.

Giả sử hàm số ||x||α||x||α là một norm bất kỳ của vector xx. Ứng với norm này, định nghĩa norm tương ứng cho ma trận AA:||A||α=maxx||Ax||α||x||α||A||α=maxx||Ax||α||x||α

chú ý rằng ma trận AA có thể không vuông và số cột của nó bằng với số chiều của xx. Như vậy, bản thân việc tính toán norm của ma trận là việc giải một bài toán tối ưu. Chú ý rằng hàm tối ưu có cả tử số và mẫu số là các norm trên vectors.

Chúng ta sẽ quan tâm nhiều hơn tới norm 2. Norm 2 của ma trận được định nghĩa là:||A||2=maxx||Ax||2||x||2   (1)||A||2=maxx||Ax||2||x||2   (1)

Nhận thấy rằng nếu xx là nghiệm của bài toán tối ưu (1)(1) thì kxkx cũng là nghiệm với kk là một số thực khác không bất kỳ. Không mất tính tổng quát, ta có thể giả sử mẫu số bằng 1. Khi đó, bài toán tối ưu (1)(1) có thể được viết dưới dạng:||A||2=max||x||2=1||Ax||2   (2)||A||2=max||x||2=1||Ax||2   (2)Nói cách khác, ta cần đi tìm xx sao cho:x=argmaxx||Ax||22 s.t.: ||x||22=1(3)x=argmaxx||Ax||22 s.t.: ||x||22=1(3)Ở đây, các norm 2 đã được bình phương lên để tránh dấu căn bậc hai. Bài toán (3)(3) có thể được giải bằng [Phương pháp nhân tử Lagrange](https://machinelearningcoban.com/2017/04/02/duality/#--phuong-phap-nhan-tu-lagrange) vì ràng buộc là một phương trình.

Lagrangian của Bài toán (3)(3) là:L(x,λ)=||Ax||22+λ(1−||x||22)L(x,λ)=||Ax||22+λ(1−||x||22)

Nghiệm của bài toán (3)(3) sẽ thoả mãn hệ phương trình:∂L∂x=2ATAx−2λx=0(4) ∂L∂λ=1−||x||22=0(5)∂L∂x=2ATAx−2λx=0(4) ∂L∂λ=1−||x||22=0(5)

Từ (4)(4) ta có:ATAx=λx    (6)ATAx=λx    (6)Điều này suy ra rằng λλ là một trị riêng của ATAATA và xx là 1 vector riêng ứng với trị riêng đó. Tiếp tục nhân hai vế của (6)(6) với xTxT vào bên trái, ta có:xTATAx=λxTx=λxTATAx=λxTx=λNhận thấy rằng vế trái chính là ||Ax||22||Ax||22 chính là hàm mục tiêu trong (3)(3). Vậy hàm mục tiêu đạt giá trị lớn nhất khi λλ đạt giá trị lớn nhất. Nói cách khác, λλ chính là trị riêng lớn nhất của ATAATA hay chính là [singular value](https://machinelearningcoban.com/2017/06/07/svd/#-nguon-goc-ten-goi-singular-value-decomposition) lớn nhất của ma trận AA.

Như vậy, norm 2 của một ma trận chính là singular value lớn nhất của ma trận đó. Và nghiệm của bài toán (3)(3) chính một là *right-singular vector* ứng với singular value đó.

Với lý luận tương tự, chúng ta có thể suy ra rằng bài toán:min||x||2=1xTATAxmin||x||2=1xTATAx

có nghiệm là vector riêng ứng với trị riêng nhỏ nhất của ATAATA. Khi đó, hàm số đạt giá trị nhỏ nhất bằng chính trị riêng nhỏ nhất này.

2.2. Biễu diễn vector trong các hệ cơ sở khác nhau

Trong không gian DD chiều , toạ độ của mỗi điểm được xác định dựa trên một hệ toạ độ nào đó. Ở các hệ toạ độ khác nhau, hiển nhiên là toạ độ của mỗi điểm cũng khác nhau.

Tập hợp các vector e1,…,eDe1,…,eD mà mỗi vector eded có đúng 1 phần tử khác 0 ở thành phần thứ dd và phần tử đó bằng 1, được gọi là hệ cơ sở đơn vị (hoặc hệ đơn vị) trong không gian DD chiều. Nếu xếp các vector ed,d=1,2,…,Ded,d=1,2,…,D theo đúng thứ tự đó, ta sẽ được ma trận đơn vị DD chiều.

Mỗi vector cột x=[x1,x2,…,xD]∈RDx=[x1,x2,…,xD]∈RD, biểu diễn của nó trong hệ đơn vị là:

x=x1e1+x2e2+⋯+xDeDx=x1e1+x2e2+⋯+xDeD

Giả sử có một hệ cơ sở khác u1,u2,…,uDu1,u2,…,uD (các vector này độc lập tuyến tính), vậy thì biểu diễn của vector xx trong hệ cơ sở mới này có dạng:

x=y1u1+y2u2+⋯+yDuD=Uyx=y1u1+y2u2+⋯+yDuD=Uy

UU là ma trận mà cột thứ dd của nó chính là vector udud. Lúc này, vector yy chính là biểu diễn của xx trong hệ cơ sở mới. Bộ các số yd,d=1,2,…,Dyd,d=1,2,…,D là duy nhất vì yy có thể tính được bằng:y=U−1x   (7)y=U−1x   (7)với chú ý rằng UU là ma trận khả nghịch vì các cột của nó độc lập tuyến tính.

Trong các ma trận đóng vai trò như hệ cơ sở UU, [các ma trận trực giao](https://machinelearningcoban.com/2017/06/07/svd/#-he-truc-giao-va-truc-chuan), tức UTU=IUTU=I, được quan tâm nhiều hơn vì nghịch đảo của chúng chính là chuyển vị của chúng:

U−1=UTU−1=UTKhi đó, yy trong (7)(7) có thể được tính một cách nhanh chóng:

y=UTxy=UTxtừ đó suy ra: yi=xTui=uTix,i=1,2,…,Dyi=xTui=uiTx,i=1,2,…,D.

Có thể nhận thấy rằng vector 00 được biểu diễn như nhau trong mọi hệ cơ sở. Hình 1 dưới đây là 1 ví dụ về việc chuyển hệ cơ sở:

|  |  |
| --- | --- |
|  | Hình 1: Chuyển đổi toạ độ trong các hệ cơ sở khác nhau. |

Việc chuyển đổi hệ cơ sở sử dụng ma trận trực giao có thể được coi như một phép xoay trục toạ độ. Nhìn theo một cách khác, đây cũng chính là một phép xoay vector dữ liệu theo chiều ngược lại.

2.3. Trace

Hàm số trace xác định trên tập các ma trận vuông được sử dụng rất nhiều trong tối ưu vì những tính chất đẹp của nó. Hàm trace trả về tổng các phần tử trên đường chéo của một ma trận vuông.

Các tính chất quan trọng của hàm trace, với giả sử rằng các ma trận trong hàm trace là vuông và các phép nhân ma trận thực hiện được:

* trace(A)=trace(AT)trace(A)=trace(AT)
* trace(kA)=ktrace(A)trace(kA)=ktrace(A) với kk là một số bất kỳ.
* trace(AB)=trace(BA)trace(AB)=trace(BA)
* ||A||2F=trace(ATA)=trace(AAT)||A||F2=trace(ATA)=trace(AAT) với AA là ma trận bất kỳ, có thể không vuông.
* trace(A)=∑Di=1λitrace(A)=∑i=1Dλi với AA là một ma trận vuông và λi,i=1,2,…,Nλi,i=1,2,…,N là toàn bộ các trị riêng của nó, có thể phức hoặc lặp. Việc chứng minh tính chất này có thể được dựa trên ma trận đặc trưng của AA và định lý Viète. Tôi xin được bỏ qua.

2.4. Kỳ vọng và ma trận hiệp phương sai

2.4.1. Dữ liệu một chiều

Cho NN giá trị x1,x2,…,xNx1,x2,…,xN. *Kỳ vọng* và *phương sai* của bộ dữ liệu này được định nghĩa là:

¯x=1NN∑n=1xn=1NX1 σ2=1NN∑n=1(xn−¯x)2x¯=1N∑n=1Nxn=1NX1 σ2=1N∑n=1N(xn−x¯)2với 1∈RN1∈RN là vector cột chứa toàn phần tử 1. Kỳ vọng đơn giản là trung bình cộng của toàn bộ các giá trị. Phương sai là trung bình cộng của bình phương khoảng cách từ mỗi điểm tới kỳ vọng. Phương sai càng nhỏ thì các điểm dữ liệu càng gần với kỳ vọng, tức các điểm dữ liệu càng giống nhau. Phương sai càng lớn thì ta nói dữ liệu càng có tính phân tán. Ví dụ về kỳ vọng và phương sai của dữ liệu một chiều có thể được thấy trong Hình 2a).

Căn bậc hai của phương sai, σσ còn được gọi là độ lệch chuẩn (standard deviation) của dữ liệu.

2.4.2. Dữ liệu nhiều chiều

Cho NN điểm dữ liệu được biểu diễn bởi các vector cột x1,…,xNx1,…,xN, khi đó, *vector kỳ vọng* và *ma trận hiệp phương sai* của toàn bộ dữ liệu được định nghĩa là:

¯x=1NN∑n=1xn S=1NN∑n=1(xn−¯x)(xn−¯x)T=1N^X^XTx¯=1N∑n=1Nxn S=1N∑n=1N(xn−x¯)(xn−x¯)T=1NX^X^TTrong đó ^XX^ được tạo bằng cách trừ mỗi cột của XX đi ¯xx¯:^xn=xn−¯xx^n=xn−x¯

Các công thức này khá tương đồng với các công thức cho dữ liệu 1 chiều phía trên. Có một vài điểm lưu ý:

* Ma trận hiệp phương sai là một ma trận đối xứng, hơn nữa, nó là một ma trận [nửa xác định dương](https://machinelearningcoban.com/2017/03/12/convexity/#positive-semidefinite).
* Mọi phần tử trên đường chéo của ma trận hiệp phương sai là các số không âm. Chúng cũng chính là phương sai của từng chiều của dữ liệu.
* Các phần tử ngoài đường chéo sij,i≠jsij,i≠j thể hiện sự tương quan giữa thành phần thứ ii và thứ jj của dữ liệu, còn được gọi là hiệp phương sai. Giá trị này có thể dương, âm hoặc bằng 0. Khi nó bằng 0, ta nói rằng hai thành phần i,ji,j trong dữ liệu là không tương quan (uncorrelated).
* Nếu ma trận hiệp phương sai là ma trận đường chéo, ta có dữ liệu hoàn toàn không tương quan giữa các chiều.

Ví dụ về dữ liệu không tương quan và tương quan được cho trong Hình 2bc).

|  |  |
| --- | --- |
| a) | b) |
| c) | Hình 2: Ví dụ về kỳ vọng và phương sai. a) Trong không gian 1 chiều. b) Không gian 2 chiều mà hai chiều không tương quan. Trong trường hợp này, ma trận hiệp phương sai là ma trận đường chéo với hai phần tử trên đường chéo là σ1,σ2σ1,σ2, đây cũng chính là hai trị riêng của ma trận hiệp phương sai và là phương sai của mỗi chiều dữ liệu. c) Dữ liệu trong không gian hai chiều có tương quan. Theo mỗi chiều, ta có thể tính được kỳ vọng và phương sai. Phương sai càng lớn thì dữ liệu trong chiều đó càng phân tán. Trong ví dụ này, dữ liệu theo chiều thứ hai phân tán nhiều hơn so so với chiều thứ nhất. |

3. Principal Component Analysis

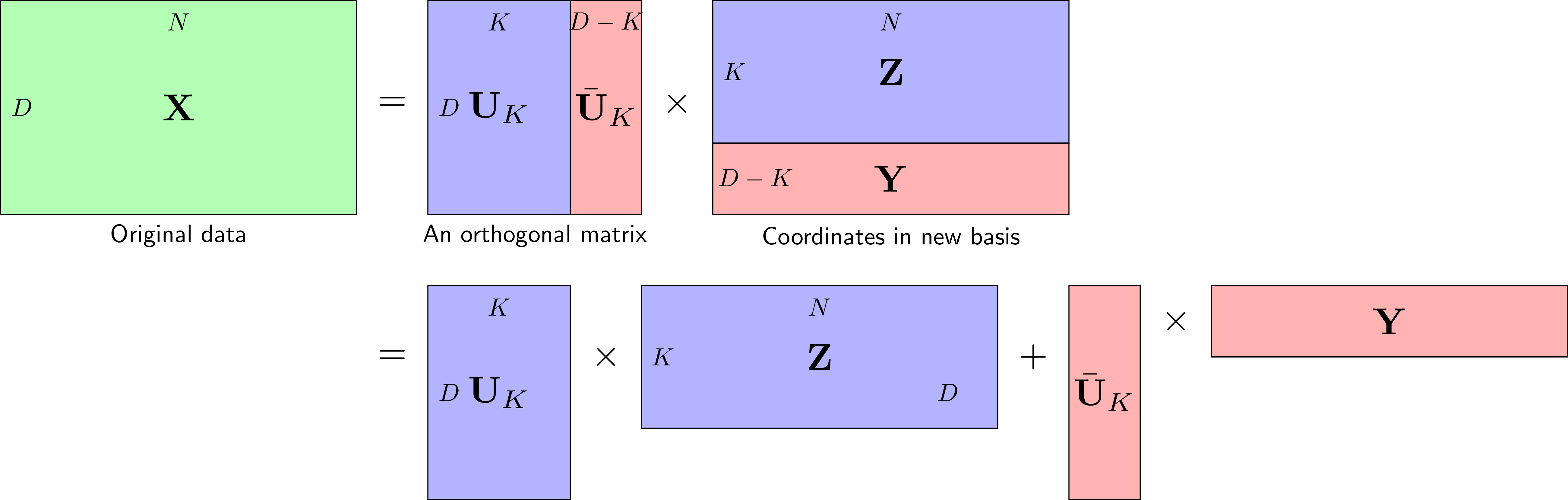
Cách đơn giản nhất để giảm chiều dữ liệu từ DD về K<DK<D là chỉ giữ lại KK phần tử *quan trọng nhất*. Tuy nhiên, việc làm này chắc chắn chưa phải tốt nhất vì chúng ta chưa biết xác định thành phần nào là quan trọng hơn. Hoặc trong trường hợp xấu nhất, lượng thông tin mà mỗi thành phần mang là như nhau, bỏ đi thành phần nào cũng dẫn đến việc mất một lượng thông tin lớn.

Tuy nhiên, nếu chúng ta có thể biểu diễn các vector dữ liệu ban đầu trong một hệ cơ sở mới mà trong hệ cơ sở mới đó, *tầm quan trọng* giữa các thành phần là khác nhau rõ rệt, thì chúng ta có thể bỏ qua những thành phần ít quan trọng nhất.

Lấy một ví dụ về việc có hai camera đặt dùng để chụp một con người, một camera đặt phía trước người và một camera đặt trên đầu. Rõ ràng là hình ảnh thu được từ camera đặt phía trước người mang nhiều thông tin hơn so với hình ảnh nhìn từ phía trên đầu. Vì vậy, bức ảnh chụp từ phía trên đầu có thể được bỏ qua mà không có quá nhiều thông tin về hình dáng của người đó bị mất.

PCA chính là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài toạ độ, phần còn lại chỉ mang một lượng nhỏ thông tin. Và để cho đơn giản trong tính toán, PCA sẽ tìm một [hệ trực chuẩn](https://machinelearningcoban.com/2017/06/07/svd/#-he-truc-giao-va-truc-chuan) để làm cơ sở mới.

Giả sử hệ cơ sở trực chuẩn mới là UU và chúng ta muốn giữ lại KK toạ độ trong hệ cơ sở mới này. Không mất tính tổng quát, giả sử đó là KK thành phần đầu tiên. Quan sát Hình 3 dưới đây:



Hình 3: Ý tưởng chính của PCA: Tìm một hệ trực chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất nằm trong KK thành phần đầu tiên.

Quan sát hình vẽ trên với cơ sở mới U=[UK,¯UK]U=[UK,U¯K] là một hệ trực chuẩn với UKUK là ma trận con tạo bởi KK cột đầu tiên của UU. Với cơ sở mới này, ma trận dữ liệu có thể được viết thành:X=UKZ+¯UKY    (8)X=UKZ+U¯KY    (8)Từ đây ta cũng suy ra:

[ZY]=[UTK¯UTK]X⇒Z=UTKX Y=¯UTKX    (9)[ZY]=[UKTU¯KT]X⇒Z=UKTX Y=U¯KTX    (9)

Mục đích của PCA là đi tìm ma trận trực giao UU sao cho phần lớn thông tin được giữ lại ở phần màu xanh UKZUKZ và phần màu đỏ ¯UKYU¯KY sẽ được lược bỏ và thay bằng một ma trận không phụ thuộc vào từng điểm dữ liệu. Nói cách khác, ta sẽ xấp xỉ YY bởi một ma trận có toàn bộ các cột là như nhau. *Chú ý rằng các cột này có thể phụ thuộc vào dữ liệu training nhưng không phụ thuộc vào dữ liệu test, các bạn sẽ thấy rõ hơn khi lập trình mà tôi sẽ trình bày trong bài tiếp theo*. Gọi mỗi cột đó là bb và có thể coi nó là bias, khi đó, ta sẽ xấp xỉ:

Y≈b1TY≈b1TTrong đó 1T∈R1×N1T∈R1×N là vector hàng có toàn bộ các phần tử bằng 1. Giả sử đã tìm được UU, ta cần tìm bb thoả mãn:b=argminb||Y−b1T||2F=argminb||¯UTKX−b1T||2Fb=argminb||Y−b1T||F2=argminb||U¯KTX−b1T||F2

Giải phương trình đạo hàm theo bb của hàm mục tiêu bằng 0:(b1T−¯UTKX)1=0⇒Nb=¯UTKX1⇒b=¯UTK¯x(b1T−U¯KTX)1=0⇒Nb=U¯KTX1⇒b=U¯KTx¯

Như vậy, **việc tính toán sẽ thuận tiện hơn nhiều nếu vector kỳ vọng**¯x=0x¯=0. Việc này có thể đạt được nếu ngay từ đầu, chúng ta trừ mỗi vector dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu. Đây chính là các bước đầu tiên của PCA.

Với giá trị bb tìm được này, dữ liệu ban đầu sẽ được xấp xỉ với:

X≈~X=UKZ+¯UK¯UTK¯x1T   (10)X≈X~=UKZ+U¯KU¯KTx¯1T   (10)

Kết hợp (8),(9),(10)(8),(9),(10) ta định nghĩa hàm mất mát chính như sau:J=1N||X−~X||2F=1N||¯UK¯UTKX−¯UK¯UTK¯x1T||2F    (11)J=1N||X−X~||F2=1N||U¯KU¯KTX−U¯KU¯KTx¯1T||F2    (11)

Chú ý rằng, nếu các cột của một ma trận VV tạo thành một hệ trực chuẩn thì với một ma trận WW bất kỳ, ta luôn có:||VW||2F=trace(WTVTVW)=trace(WTW)=||W||2F||VW||F2=trace(WTVTVW)=trace(WTW)=||W||F2

Vì vậy hàm mất mát trong (11)(11) có thể viết lại thành:J=1N||¯UTK(X−¯x1)T||2F=1N||¯UTK^X||2F =1N||^XT¯UK||2F=1ND∑i=K+1||^XTui||22 =1ND∑i=K+1uTi^X^XTui =D∑i=K+1uTiSui(12)J=1N||U¯KT(X−x¯1)T||F2=1N||U¯KTX^||F2 =1N||X^TU¯K||F2=1N∑i=K+1D||X^Tui||22 =1N∑i=K+1DuiTX^X^Tui =∑i=K+1DuiTSui(12)

Với ^X=X−¯x1TX^=X−x¯1T là dữ liệu đã chuẩn hoá và với SS là ma trận hiệp phương sai của dữ liệu. Ta gọi ma trận này ^XX^ là *zero-corrected data* hoặc *dữ liệu đã được chuẩn hoá*. Có thể nhận thấy ^xn=xn−¯xx^n=xn−x¯.

Công việc còn lại là tìm các uiui để mất mát là nhỏ nhất. Trước hết, chúng ta có một nhận xét thú vị. Nhắc lại định nghĩa ma trận hiệp phương sai S=1N^XT^XS=1NX^TX^. Với ma trận UU trực giao bất kỳ, thay K=0K=0 vào (12)(12) ta có:

L=D∑i=1uTiSui=1N||^XTU||2F =1Ntrace(^XTUUT^X)(12) =1Ntrace(^XT^X)(13) =1Ntrace(^X^XT)(14) =trace(S)=D∑i=1λi(16)L=∑i=1DuiTSui=1N||X^TU||F2 =1Ntrace(X^TUUTX^)(12) =1Ntrace(X^TX^)(13) =1Ntrace(X^X^T)(14) =trace(S)=∑i=1Dλi(16)

Với λ1≥λ2≥⋯≥λD≥0λ1≥λ2≥⋯≥λD≥0 là các trị riêng của ma trận nửa xác định dương SS. Chú ý rằng các trị riêng này là thực và không âm.

**Như vậy**LL**không phụ thuộc vào cách chọn ma trận trực giao**UU và bằng tổng các phần tử trên đường chéo của SS. Nói cách khác, LL chính là tổng của các phương sai theo từng thành phần của dữ liệu ban đầu.

Vì vậy, việc tối thiểu hàm mất mát JJ được cho bởi (13)(13) tương đương với việc tối đa:F=L−J=K∑i=1uiSuTiF=L−J=∑i=1KuiSuiT

**Định lý 1:** FF đạt giá trị lớn nhất bằng ∑Ki=1λi∑i=1Kλi khi uiui là các vector riêng có norm 2 bằng 1 ứng với các trị riêng này. Tất nhiên, chúng ta không quên điều kiện trực giao giữa các uiui.

Chú ý rằng λi,i=1,…,Kλi,i=1,…,K chính là KK trị riêng lớn nhất của ma trận hiệp phương sai SS. Trị riêng lớn nhất λ1λ1 của ma trận này còn được gọi là *Thành phần chính thứ nhất* (First Principal Component), trị riêng thứ hai λ2λ2 còn được gọi là *Thành phần chính thứ hai*, etc. Chính vì vậy, phương pháp này có tên gọi là *Phân tích thành phần chính* - Principal Component Analysis. Ta chỉ giữ lại KK thành phần chính của dữ liệu khi muốn giảm số chiều dữ liệu. Để có cái nhìn trực quan hơn, chúng ta cùng theo dõi Hình dưới đây:

|  |  |
| --- | --- |
|  | Hình 4: PCA dưới góc nhìn Thống kê. PCA có thể được coi là phương pháp đi tìm một hệ cơ sở trực chuẩn đóng vai trò một phép xoay, sao cho trong hệ cơ sở mới này, phương sai theo một số chiều nào đó là rất nhỏ, và ta có thể bỏ qua. |

Trong không gian ban đầu với các vector cơ sở màu đen e1,e2e1,e2, phương sai theo mỗi chiều dữ liệu đều lớn. Trong không gian mới với các vector cơ sở màu đỏ u1,u2u1,u2, phương sai theo chiều thứ hai ^σ2σ^2 rất nhỏ so với ^σ1σ^1. Điều này nghĩa là khi chiếu dữ liệu lên u2u2 ta được các điểm rất gần nhau và gần với kỳ vọng theo chiều đó. Trong trường hợp này, kỳ vọng theo mọi chiều bằng 0 nên ta có thể thay thế toạ độ theo chiều u2u2 bằng 0. Rõ ràng là nếu dữ liệu có phương sai càng nhỏ theo một chiều nào đó thì khi xấp xỉ chiều đó bằng một hằng số, sai số xảy ra càng nhỏ. PCA thực chất là đi tìm một phép xoay tương ứng với một ma trận trực giao sao cho trong hệ toạ độ mới, tồn tại các chiều có phương sai nhỏ mà ta có thể bỏ qua; ta chỉ cần giữ lại các chiều/thành phần khác quan trọng hơn. Như đã chứng minh ở trên, tổng phương sai theo mọi chiều trong hệ cơ sở nào cũng là như nhau và bằng tổng các trị riêng của ma trận hiệp phương sai. Vì vậy, PCA còn được coi là phương pháp giảm số chiều dữ liệu mà giữ được tổng phương sai còn lại là lớn nhất.

Tôi sẽ bỏ qua phần chứng minh của Định lý 1. Tuy nhiên, cũng nêu một vài ý để bạn đọc có thể hình dung:

Khi K=1K=1. Ta cần giải bài toán:maxu1uT1Su1 s.t.:||u1||2=1maxu1u1TSu1 s.t.:||u1||2=1

Như đã đề cập ở phía trên, hàm mục tiêu đạt giá trị lớn nhất bằng λ1λ1 khi u1u1 là một vector riêng của ma trận hiệp phương sai SS tương ứng với trị riêng λ1λ1. Vậy định lý đúng với K=1K=1

Giả sử u1u1 đã là vector riêng ứng với trị riêng lớn nhất của SS thế thì nghiệm u2u2 của bài toán tối ưu:maxu2uT2Su2 s.t.||u2||2=1(21) uT2u1=0maxu2u2TSu2 s.t.||u2||2=1(21) u2Tu1=0là một vector riêng của SS ứng với trị riêng lớn thứ hai λ2λ2 của nó. Chú ý rằng λ2λ2 có thể bằng λ1λ1 nếu không gian riêng ứng với λ1λ1 có số rank lớn hơn 1.

Nhận định này có thể được chứng minh bằng phương pháp nhân tử Lagrange. Thật vậy, Lagrangian của bài toán (21)(21) là:L(u2,ν1,ν2)=uT2Su2+ν1uT1u2+ν2(1−uT2u2)L(u2,ν1,ν2)=u2TSu2+ν1u1Tu2+ν2(1−u2Tu2)

Ta cần giải hệ phương trình đạo hàm của LL theo từng biến bằng 0:∂L∂u2=2Su2+ν1u1−2ν2u2=0(22) ∂L∂ν1=uT1u2=0(23) ∂L∂ν2=1−uT2u2=0(24) ∂L∂u2=2Su2+ν1u1−2ν2u2=0(22) ∂L∂ν1=u1Tu2=0(23) ∂L∂ν2=1−u2Tu2=0(24)

Nhân cả hai vế của (22)(22) với uT1u1T vào bên trái ta có:2uT1Su2+ν1=02u1TSu2+ν1=0Vì Su1=λ1u1⇒uT1Su2=λ1uT1u2=0Su1=λ1u1⇒u1TSu2=λ1u1Tu2=0. Từ đó suy ra ν1=0ν1=0 và (22)(22) lúc này tương đương với:Su2=ν2u2⇒uT2Su2=ν2Su2=ν2u2⇒u2TSu2=ν2Vậy u2u2 là một vector riêng của SS ứng với ν2ν2. Và để hàm mục tiêu đạt giá trị lớn nhất, ν2ν2 cần càng lớn càng tốt. Điều này dẫn đến ν2ν2 phải là trị riêng thứ hai của SS.

Lập luận tương tự, ta có thể chứng minh được: Nếu ui,i=1,2,…,k−1ui,i=1,2,…,k−1 là các vector riêng ứng với trị riêng lớn thứ ii của ma trận nửa xác định dương SS, hơn nữa, k−1k−1 vector riêng này tạo thành một hệ trực chuẩn, thế thì:

maxukuTkSuk s.t.uTkuk=1; uTkui=1,i=1,…,k−1maxukukTSuk s.t.ukTuk=1; ukTui=1,i=1,…,k−1bằng đúng với trị riêng tiếp theo λkλk tại ukuk là vector riêng ứng với trị riêng này.

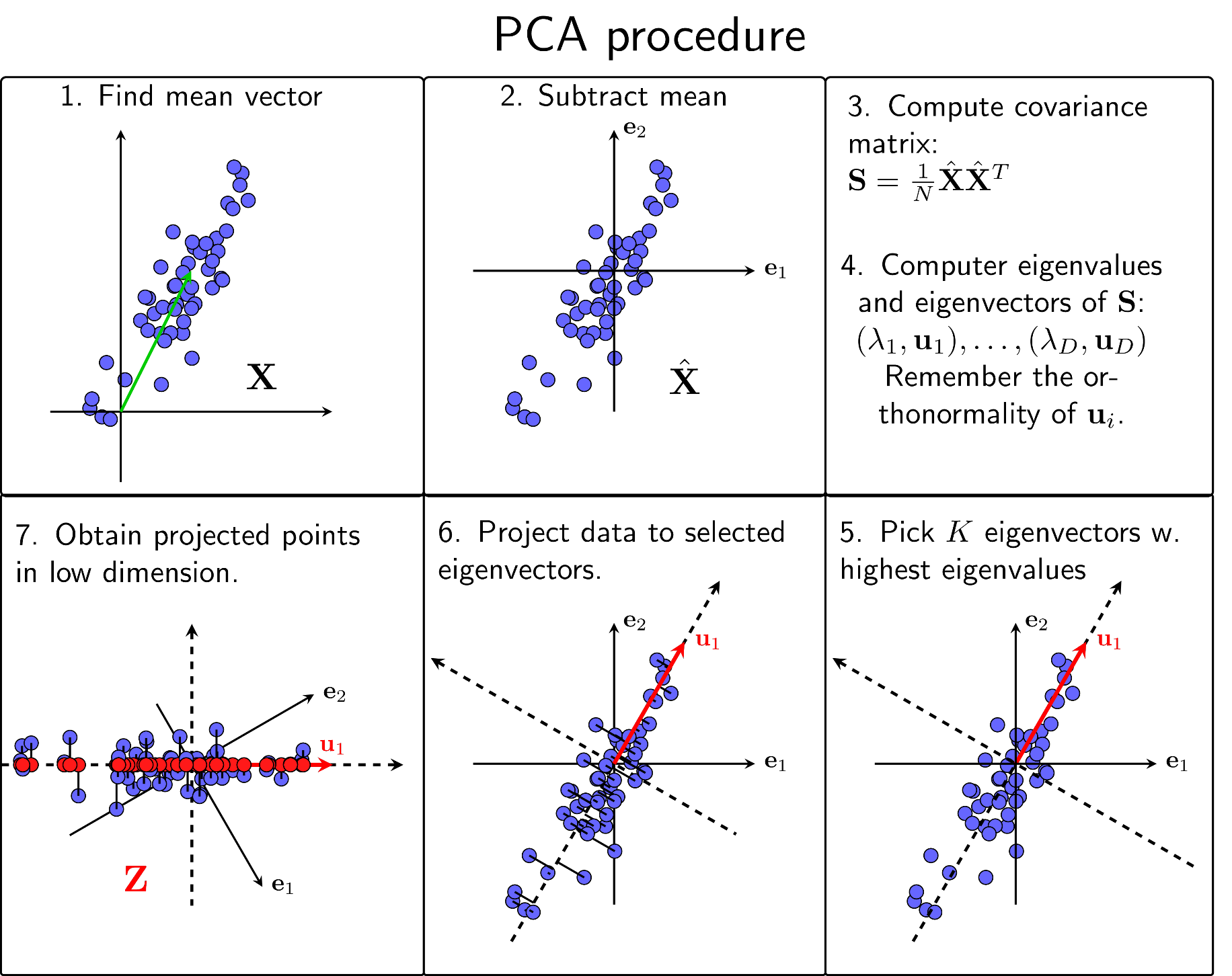
4. Các bước thực hiện PCA

Từ các suy luận phía trên, ta có thể tóm tắt lại các bước trong PCA như sau:

1. Tính vector kỳ vọng của toàn bộ dữ liệu:¯x=1NN∑n=1xnx¯=1N∑n=1Nxn
2. Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:^xn=xn−¯xx^n=xn−x¯
3. Tính ma trận hiệp phương sai:S=1N^X^XTS=1NX^X^T
4. Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
5. Chọn KK vector riêng ứng với KK trị riêng lớn nhất để xây dựng ma trận UKUK có các cột tạo thành một hệ trực giao. KK vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con *gần* với phân bố của dữ liệu ban đầu đã chuẩn hoá.
6. Chiếu dữ liệu ban đầu đã chuẩn hoá ^XX^ xuống không gian con tìm được.
7. Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới.Z=UTK^XZ=UKTX^

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:x≈UKZ+¯xx≈UKZ+x¯

Các bước thực hiện PCA có thể được xem trong Hình dưới đây:



Hình 5: Các bước thực hiện PCA.