MIT OCW Hesapsal Bilim 18.085 Ders 3

Konumuz $Au = b$ sistemini cozmek. Bu cozum icin Python'da `linalg.solve` cagrisi var. Mesela

```
import numpy as np
import scipy.linalg
A = [[2,3,4],[5,5,6],[7,7,7]]
b = [1,2,3]
u = scipy.linalg.solve(A, b)
print u
```

`linalg.solve` cagrisi Matlab'de \ cagrisinin karsiligi, oradaki kullanim u = A \ b seklinde.

Eger elimizde ikinci bir `c` vektoru var ise, ve esitligin sag tarafinda `b` sonrasi onu kullanmak istiyorsak ayri ayri `solve` komutlarina gerek yoktur. Her iki vektoru birbirine ekleyerek, `solve`'u toplu halde cagirabiliriz, bu performans acisindan daha iyi olur.

```
c = [2,3,8]
bc = np.vstack((b,c)).T
u = scipy.linalg.solve(A,   bc)
```

Python `vstack` komutu iki matrisi ust uste koymak icin kullanilir.

Her iki cozum beraber olarak geri gelecektir. Bu niye daha hizli? Cunku Python'un cozucusu daha esitligin sag tarafina bile gelmeden sadece `A`'ya bakarak bir suru islem gerceklestiriyor, eliminasyon yaparak `A`'yi ucgensel hale getirmek gibi. Bu tur islemleri gereksiz kere iki kere yapmak pahali olurdu.

Soru: matematiksel olarak u'yu bulmak

$Au = b$

$u = A^{-1}b$

demektir. Peki Python bu hesap icin gercekten A^{-1}'i hesaplar mi?

Hayir. Cunku buyuk problemler icin matris tersini hesaplamak oldukca pahalidir. Ayrica A matrisi zaten ucgensel capraz (tridiagonal) bir halde olabilir, ve cevap zaten hazir haldedir, bu noktada ters alma islemi gereksiz olurdu.

Biraz zihin egzersizi yapalim. Eger soyle bir komut kullansam ne elde ederim (ki `I` matrisi birim matrisi) ?

```
solve(A, I)
```

Cevap, tabii ki A'nin tersini elde ederim, yani A^{-1} cunku $AA^{-1} = I$, sag tarafta birim matrisi var ise cozum sadece A^{-1} olabilir.

$$A \left[\begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \right] = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right]$$

Bu probleme bakmanin degisik bir yolu: sag taraftaki birim matrisi icindeki [1 0 0] gibi degerler icindeki 1 degerlerini birer ziplama (impulse) ani gibi gormek, sanki elimizde bir duz [0 0 .. 0] bir veri var, icinde tek ziplama olan yer orasi, ve bu [1 0 0], [0 1 0], .. icinde tek ziplama olan veriler "islenerek" bize u_1, u_2, .. gibi sonuclari veriyorlar.

Elle A'nin tersini bulmak icin ne yapardik? Bir blok matrisi yaratirdik, `[A I]`, yani 3x3 ve 3x3 iki matrisi yanyana koyup 3x6 boyutunda yeni bir matris elde ederdik, ve bu matriste A uzerinde eliminasyon, pivotlari sifirlama gibi numaralari kullanarak onu birim matrise cevirirdik, bu arada ayni operasyonlari tabii ki I uzerinde uygulardik. En sonunda A birim olunca I A^{-1}'e donusmus olurdu!

Simdi biraz buyuk resme bakalim.

Lineer cebirin 4 buyuk problemi Python komutlari ile beraber sunlardir:

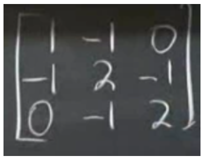
Eliminasyon, `scipy.linalg.lu(A)` $A = LU$

Dikeylestirme / ortogonalastirme (orthogonalization), `scipy.linalg.qr(A)`, $A = QR$

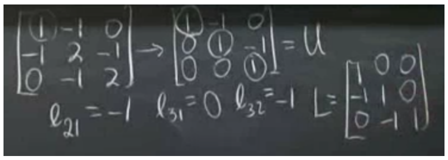
Ozdegerler (eigenvalues), `scipy.linalg.eig(A)` $A = SAS^{-1}$

Tekil degerler (singular values), `scipy.linalg.svd(A)` $A = U\Sigma V^T$

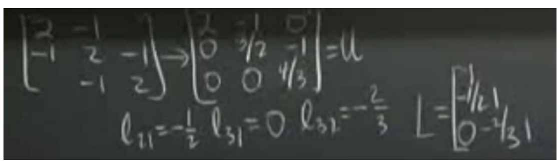
Eliminasyon ne yapar? Dikkat edersek aslinda bu islemin bir alt ucgensel (lower triangular) matris (L) ve bir tane de ust ucgensel matris (U) ortaya cikardigini goruruz. Simdi alttaki matris uzerinde eliminasyon yapalim ve bu arada tersini de bulmus olalim.



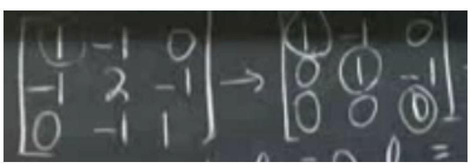
Eliminasyon islemlerini yapalim.



$l_{21} = -1$ yapilan ilk islemi kodluyor, 1. satiri -1 ile carp ve 2. satirdan cikart anlamina geliyor. Digerlerini de sirasiyle goruyoruz ve bu islemlerin sonucunda ust ucgensel matris U'yu elde ediyoruz. Tum l degerlerini bir araya koyup L'yi elde edebiliriz. Bir tane daha yapalim:

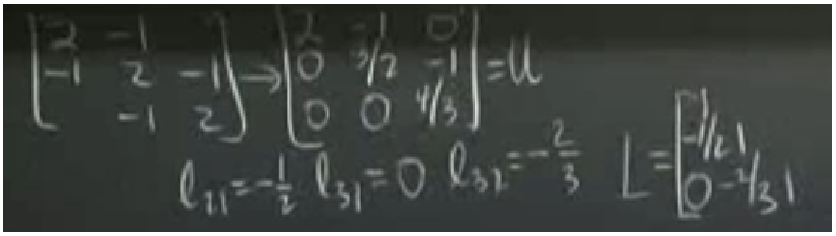


Eger tekil (singular) bir matris uzerinde eliminasyon yapsak, bu islemi nasil etkilerdi?

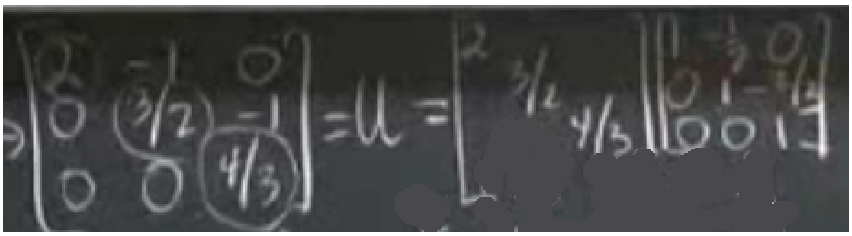


Yani bu durumda 3 tane pivot elde edemezdik, sag alt kosedeki deger eliminasyon sirasinda 0 olurdu, ve sag matris, aynen sol matris gibi, tekil olurdu. Bu isimize yaramazdi.

Su probleme donelim:



Burada ilk matris simetrik idi, ama L ve U matrisi artik simetrik degil. Simetriyi geri getirebilir miyiz? U icinden sadece caprazlari cekip cikartalim



Boylece caprazinda [2 3/2 4/3] olan bir matris elde ettik. Peki bu matrisin carptigi (onun hemen saginda) icinden caprazlari cekip cikardigimiz matristen geri kalanlar tanidik geliyor mu? Evet, bu matris te L^T'e esit. Demek ki $LU = LDL^T$ gibi bir ifade mumkun.

Biliyoruz ki

$K = LDL^T$

ifadesinde K her zaman simetriktir. Ters yonden soylersek, herhangi bir simetrik K matrisini alip eliminasyon yaparsam ve L ve D elde edince, L^T ile carpabilirim.

Peki sunu ispat edebilir miyiz? Herhangi bir L ve capraz D var ise, LDL^T her zaman simetrik midir? Bir matrisin simetrik olmasi demek kendi devrigine (transpose) esit olmasi demektir. Yani

$K = LDL^T$

$K^T = (LDL^T)^T$

Devrigi alinca parantez icindeki carpimlarin sirasi degisir.

$= (L^T)^T D^T L^T$

$D^T = D$ cunku D zaten capraz bir matris, onemli tum degerleri caprazinda ve devrik islemi bu durumu degistirmiyor. O zaman

$$= LDL^T$$

Tekrar basladigimiz noktaya donduk. Demek ki basladigimiz matris simetriktir. Ispat tamamlandi.

Genele donelim: $A^T A$'nin mesela karesel oldugunu biliyorduk (nxm ile mxn carpilinca nxn boyutu elde edilir). Simdi bunun uzerine simetrik oldugunu da artik biliyoruz, ustte ispatladik.

Kural: Simetrik matrislerin tersi (inverse) de simetriktir. O zaman K^{-1} de simetriktir.