Inverse of a Partitioned Matrix:

There is no general methodology for partitioned matrices. Here, we get the flavor of how exploiting partitioning might be a good idea.

Take the special case $A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline \emptyset & A_{22} \end{array}\right]$ $\quad A_{22} \in \mathbb{R}^{q\times q}$, $A_{11} \in \mathbb{R}^{p\times p}$

Goal: find A^{-1} (assume A^{-1} exists):

Using row echelon operations, find a matrix B s.t.

$$\begin{bmatrix} A_{11} & A_{12} \\ \emptyset & A_{22} \end{bmatrix}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} I_p & \emptyset \\ \emptyset & I_q \end{bmatrix}$$

Perform matrix multiplication:

$$② \left\{ \begin{array}{ll} A_{11}B_{11} + A_{12}B_{21} = I_p & (\dagger) \\ A_{22}B_{21} = \emptyset \Rightarrow B_{21} = \emptyset \text{ since } A_{22} \neq \emptyset. \end{array} \right.$$

$$① \left\{ \begin{array}{ll} A_{11}B_{12} + A_{12}B_{22} = \emptyset & (\ast) \\ A_{22}B_{22} = I_q \Rightarrow B_{22} = A_{22}^{-1} \in \mathbb{R}^{q\times q} \end{array} \right.$$

That $B_{22} = A_{22}^{-1}$ does not tell us what B_{22} is. We still will have to solve for it.

Use (#) in (1)

$$A_{11} B_{12} + A_{12} B_{22} = \emptyset$$

$$\text{or } A_{11} B_{12} = -A_{12} B_{22} = -A_{12} A_{22}^{-1}$$

hence $\boxed{B_{12} = -A_{11}^{-1} A_{12} A_{22}^{-1}}$

Again, we need to find A_{11}^{-1} as well.

$$\therefore A^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1} A_{12} A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{bmatrix}$$

What's gained? by direct means computing A^{-1} is $O(N^3)$ where $N = p+q$. Computing A_{11}^{-1} is $O(p^3)$ and $A_{22}^{-1} = O(q^3)$. Much cheaper. Compare $O((p+q)^3)$ to $O(p^3) + O(q^3)$. We have not talked about numerical stability, but one can gain numerical stability (in some cases).

MATRIX FACTORIZATION

There are all kinds of factorizations:

LU , Cholesky, QR, SVD, ILU, etc

LU Factorization

As we'll see,

Can be related to matrix multiplication

by matrices that effect row reduction:

Elementary Matrices : (simple) matrices

that effect row operations on another matrix

Elementary Matrix

ex) $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ & $E_1 = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix}$

$$E_1 A = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ -4a+c & -4b+d \end{bmatrix}$$

So E_1 applied to A did the following:

added : -4(row1) to row2

left unchanged : row1 = row1

//

So the many row operations, applied to some matrix A, can be written as the LEFT product $E_p E_{p-1} E_{p-2} \cdots E_2 E_1 A$

[If $FE = I \Rightarrow F$ is the "reverse" of E.

Not every elementary matrix has an inverse.]

However, the product of these could:

An example of find A^{-1}, here $A \in \mathbb{R}^{n\times n}$

If $(E_p E_{p-1} \cdots E_1) A = I_n$

then $E_p E_{p-1} \cdots E_1 = A^{-1}$ (if it exists)

LU Factorization

In matlab

lu(A) gives the LU factorization of the matrix

$A \in \mathbb{R}^{m\times n}$

$A =$

$L \in \mathbb{R}^{m\times m}$ $U \in \mathbb{R}^{m\times n}$

$$A = \underbrace{\begin{bmatrix} 1 & & 0 \\ & \ddots & \\ x & & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} \times & & x \\ & \ddots & \\ 0 & & \times \end{bmatrix}}_{U} \quad \mathbb{R}^{n\times n}$$

Why is this a good idea?

ex) (#) $A\underset{\sim}{x} = \underline{b}$, b is known. Want to solve for unknown $\underset{\sim}{x}$. Here $A \in \mathbb{R}^{n\times n}$, Then

$\underset{\sim}{x} = A^{-1}\underset{\sim}{b}$, as we know.

Use LU factorization

let $A = LU$

replace into (#)

$$LU\underset{\sim}{x} = \underline{b}$$

$$L(U\underset{\sim}{x}) = b \Rightarrow \begin{cases} L\underset{\sim}{z} = \underset{\sim}{b} & (*) \\ U\underset{\sim}{x} = \underset{\sim}{z} & (\$) \end{cases}$$

$$(*) \quad \begin{pmatrix} 1 & & & & 0 \\ & 1 & & & \\ & & 1 & & \\ & x & & 1 & \\ & & & & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

Easily find $\underset{\sim}{z}$ ($O(n)$ ops)

use it in ($\ddagger$):

$$\begin{pmatrix} \diagdown & X \\ 0 & \diagdown \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$

to solve for $\underset{\sim}{x}$ in $O(n)$ ops.

So, once we have done the LU factorization (which generally takes $O(N^3)$ operations), then we can solve problems of the form

$$A \underset{\sim}{x}_n = \underset{\sim}{b}_n$$

$n = 1, 2, \ldots$

Here A is known, and $\underset{\sim}{b}_n$ is known.
You factor $A = LU$ once, then solving for each x_n is cheap.