STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2019
LECTURE 13

1. determinants and inverses with Schur complement

- recall the block LU decomposition
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & S \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}$$
where
$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$
is the Schur complement
- this gives us a nice way to evaluate determinant of block matrix
$$\det(A) = \det(A_{11})\det(S)$$
- it also gives us a formula for the inverse of block matrix
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{bmatrix}$$
- the trick to derive this expression is to consider
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$
and try to express
$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$
and in which case $B = A^{-1}$
- we already have
$$(A_{22} - A_{21}A_{11}^{-1}A_{12})\mathbf{x}_2 = \mathbf{b}_2 - A_{21}A_{11}^{-1}\mathbf{b}_1 \tag{1.1}$$
from the last lecture which expresses $\mathbf{x}_2$ in terms of $\mathbf{b}_1$ and $\mathbf{b}_2$
- we need something similar for $\mathbf{x}_1$ and so we plug (1.1) back into
$$\mathbf{x}_1 = A_{11}^{-1}(\mathbf{b}_1 - A_{12}\mathbf{x}_2)$$
from our last lecture, which gives us
$$\mathbf{x}_1 = (A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1})\mathbf{b}_1 - A_{11}^{-1}A_{12}S^{-1}\mathbf{b}_2 \tag{1.2}$$
- now we just write (1.1) and (1.2) in block form
$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$
which yields the required formula

Date: November 11, 2019, version 1.0. Comments, bug reports: `lekheng@galton.uchicago.edu`.

2. Rank-1 Updating

- suppose that we have solved the problem $A\mathbf{x} = \mathbf{b}$ and we wish to solve the perturbed problem
$$(A + \mathbf{u}\mathbf{v}^\mathsf{T})\mathbf{y} = \mathbf{b}$$
- such a perturbation is called a *rank-one update* of A, since the matrix $\mathbf{u}\mathbf{v}^\mathsf{T}$ has rank 1 (unless $\mathbf{u}$ or $\mathbf{v}$ is zero)
- as an example, we might find that there was an error in the element a_{11} and we update it with the value $\bar{a}_{11}$
- we can accomplish this update by setting
$$\bar{A} = A + (\bar{a}_{11} - a_{11})\mathbf{e}_1\mathbf{e}_1^\mathsf{T}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
- for a general rank-one update, we can use the *Sherman–Morrison formula*, which we will derive here
- multiplying through the equation $(A + \mathbf{u}\mathbf{v}^\mathsf{T})\mathbf{y} = \mathbf{b}$ by A^{-1} yields
$$(I + A^{-1}\mathbf{u}\mathbf{v}^\mathsf{T})\mathbf{y} = A^{-1}\mathbf{b} = \mathbf{x}$$
- we therefore need to find $(I + \mathbf{w}\mathbf{v}^\mathsf{T})^{-1}$ where $\mathbf{w} = A^{-1}\mathbf{u}$
- we assume that $(I + \mathbf{w}\mathbf{v}^\mathsf{T})^{-1}$ is a matrix of the form $(I + \sigma\mathbf{w}\mathbf{v}^\mathsf{T})$ where σ is some constant
- from the relationship
$$(I + \mathbf{w}\mathbf{v}^\mathsf{T})(I + \sigma\mathbf{w}\mathbf{v}^\mathsf{T}) = I$$
we obtain
$$\sigma\mathbf{w}\mathbf{v}^\mathsf{T} + \mathbf{w}\mathbf{v}^\mathsf{T} + \sigma\mathbf{w}\mathbf{v}^\mathsf{T}\mathbf{w}\mathbf{v}^\mathsf{T} = 0$$
- however, the quantity $\mathbf{v}^\mathsf{T}\mathbf{w}$ is a scalar, so this simplifies to
$$(\sigma + 1 + \sigma\mathbf{v}^\mathsf{T}\mathbf{w})\mathbf{w}\mathbf{v}^\mathsf{T} = 0$$
which yields
$$\sigma = -\frac{1}{1 + \mathbf{v}^\mathsf{T}\mathbf{w}}$$
- it follows that the solution $\mathbf{y}$ to the perturbed problem is given by
$$\mathbf{y} = (I + \sigma\mathbf{w}\mathbf{v}^\mathsf{T})\mathbf{x} = \mathbf{x} + \sigma(\mathbf{v}^\mathsf{T}\mathbf{x})\mathbf{w}$$
and the perturbed inverse is given by
$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^\mathsf{T})^{-1} &= (I + A^{-1}\mathbf{u}\mathbf{v}^\mathsf{T})^{-1}A^{-1} \\ &= \left(I - \frac{1}{1 + \mathbf{v}^\mathsf{T}\mathbf{w}}\mathbf{w}\mathbf{v}^\mathsf{T}\right)A^{-1} \\ &= A^{-1} - \frac{1}{1 + \mathbf{v}^\mathsf{T}A^{-1}\mathbf{u}}A^{-1}\mathbf{u}\mathbf{v}^\mathsf{T}A^{-1} \qquad (2.1)\end{aligned}$$
which is the Sherman–Morrison formula
- an efficient algorithm for solving the perturbed problem $(A + \mathbf{u}\mathbf{v}^\mathsf{T})\mathbf{y} = \mathbf{b}$ can therefore proceed as follows:
  - solve $A\mathbf{x} = \mathbf{b}$
  - solve $A\mathbf{w} = \mathbf{u}$
  - compute $\sigma = -1/(1 + \mathbf{v}^\mathsf{T}\mathbf{w})$
  - compute $\mathbf{y} = \mathbf{x} + \sigma(\mathbf{v}^\mathsf{T}\mathbf{x})\mathbf{w}$

- note that we already have the solution to $A\mathbf{x} = \mathbf{b}$ but we have to solve another system $A\mathbf{w} = \mathbf{u}$
- so how is this better than simply solving $(A + \mathbf{u}\mathbf{v}^{\mathsf{T}})\mathbf{y} = \mathbf{b}$?
- the answer is that if we have LU factorization of A, then solving $A\mathbf{w} = \mathbf{u}$ requires two back solves, which takes $O(n^2)$ operations whereas solving $(A + \mathbf{u}\mathbf{v}^{\mathsf{T}})\mathbf{y} = \mathbf{b}$ from scratch would require $O(n^3)$ operations
- note that this also works if we have the QR or any other factorizations of A that facilitate solving linear equations involving A
- an alternative approach is to note that

$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^{\mathsf{T}})^{-1} &= [A(I + A^{-1}\mathbf{u}\mathbf{v}^{\mathsf{T}})]^{-1} \\ &= (I + \sigma A^{-1}\mathbf{u}\mathbf{v}^{\mathsf{T}})A^{-1} \\ &= A^{-1} + \sigma A^{-1}\mathbf{u}\mathbf{v}^{\mathsf{T}}A^{-1}\end{aligned}$$

which yields

$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^{\mathsf{T}})^{-1}\mathbf{b} &= A^{-1}(I + \sigma\mathbf{u}\mathbf{v}^{\mathsf{T}}A^{-1})\mathbf{b} \\ &= A^{-1}(\mathbf{b} + \sigma(\mathbf{v}^{\mathsf{T}}A^{-1}\mathbf{b})\mathbf{u})\end{aligned}$$

and therefore we can solve $(A + \mathbf{u}\mathbf{v}^{\mathsf{T}})\mathbf{y} = \mathbf{b}$ by solving a problem of the form $A\mathbf{x} = \mathbf{b}$ where the right-hand side $\mathbf{b}$ is perturbed

3. RANK-r UPDATE

- what we have in the previous section can be generalized by repeated application of the same technique

$$A + \mathbf{u}_1\mathbf{v}_1^{\mathsf{T}} + \cdots + \mathbf{u}_r\mathbf{v}_r^{\mathsf{T}} = A + UV^{\mathsf{T}} \tag{3.1}$$

where $U = [\mathbf{u}_1, \dots, \mathbf{u}_r], V = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{n\times r}$
- (3.1) is called a *rank-r update* of A
- this is useful if, for example, r entries of A are modified, requiring us to obtain the solution of $(A + UV^{\mathsf{T}})\mathbf{x} = \mathbf{b}$ from the original solution $A\mathbf{x} = \mathbf{b}$
- as we will see this method works best when $r \ll n$
- the notion of rank-r update is very much related to that of Schur complement
- if we introduce new variables $\mathbf{y} = C\mathbf{x}$, then

$$(A + BC)\mathbf{x} = \mathbf{b}$$

can be written as

$$\begin{cases} A\mathbf{x} + B\mathbf{y} = \mathbf{b} \\ \mathbf{y} = C\mathbf{x} \end{cases} \tag{3.2}$$

or equivalently

$$\begin{bmatrix} A & B \\ C & -I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

- in other words $A + BC$ is the Schur complement of $-I$ in $\begin{bmatrix} A & B \\ C & -I \end{bmatrix}$
- we now derive a generalization of the Sherman–Morrison formula (2.1) by solving (3.2)
- plug $\mathbf{x} = A^{-1}(\mathbf{b} - B\mathbf{y})$ into $\mathbf{y} = C\mathbf{x}$ to get

$$(I + CA^{-1}B)\mathbf{y} = CA^{-1}\mathbf{b}$$

and plug the expression $\mathbf{y} = (I + CA^{-1}B)^{-1}CA^{-1}\mathbf{b}$ back into $\mathbf{x} = A^{-1}(\mathbf{b} - B\mathbf{y})$ to get

$$\mathbf{x} = [A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}]\mathbf{b}$$

- note that it is inevitable that we will have to solve a linear system involving the coefficient matrix
$$I + CA^{-1}B \in \mathbb{R}^{r\times r}$$
but when r is small, which is usually the case, this is much easier than solving a linear system with coefficient matrix
$$A + BC \in \mathbb{R}^{n\times n}$$
- since $\mathbf{b}$ is arbitrary, this must mean that
$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1} \tag{3.3}$$
- this is called the *Sherman–Woodbury–Morrison formula* and is useful for find rank-r updates of solutions to $A\mathbf{x} = \mathbf{b}$
- a word of caution: both (2.1) and (3.3) should not be used for computing explicit inverse (which is a bad idea in the first place) because they are numerically unreliable

4. WHY ITERATIVE METHODS

- if we have a linear system $A\mathbf{x} = \mathbf{b}$ where A is very, very large but is either sparse or structured (e.g., banded, Toeplitz, banded plus low-rank, semiseparable, Hierarchical, etc), the easiest way to exploit this is to use *iterative methods*
- these are methods that construct a sequence of vectors $\mathbf{x}^{(k)}$ so that $\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x} = A^{-1}\mathbf{b}$
- we shall focus on solving linear systems but there are also iterative methods for least squares problems, eigenvalue problems, singular value problems, etc — in fact for the last two, there are only iterative methods
- one big advantage of iterative methods is that we can control how accurate we want our solution, for example, if we want our solution to be ε-accurate (whether relative or absolute), then in principle we can stop as soon as
$$\|\mathbf{x}^{(k)} - \mathbf{x}\| < \varepsilon \quad \text{or} \quad \frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|} < \varepsilon \tag{4.1}$$
- if, say, $n = 10,000$ but it takes only $k = 5$ iterations to reach our desired level of accuracy, then we have saved a lot of computations — direct methods like LU, QR, Cholesky, etc, do not allow this
- in practice of course we do not know $\mathbf{x} = A^{-1}\mathbf{b}$ and it might appear that we can't use forward errors like those in (4.1) to control accuracy but we will see later that we don't need to know $\mathbf{x}$ to gurantee (4.1)
- usually iterative methods converge in the limit to the solution but there are iterative methods that actually converge in finitely many steps
- for example, many *Krylov subspace methods* converge in k steps where k = number of distinct nonzero eigenvalues of A:
  - conjugate gradient (CG) method for symmetric positive definite A
  - minimal residual (MINRES) method for symmetric A
  - general minimial resitual (GMRES) method for general A
- there are three classes of iterative methods for $A\mathbf{x} = \mathbf{b}$
  - *splitting methods*: decompose A into the sum of two matrices
  $$A = M - N$$
  where M is easy to invert and then do
  $$M\mathbf{x}^{(k)} = N\mathbf{x}^{(k-1)} + \mathbf{b}$$
  these are also known as *one-step stationary methods*

  - *semi-iterative methods*: generate
    $$\mathbf{y}^{(k)} = B\mathbf{y}^{(k-1)} + \mathbf{c}$$
    for suitable B and $\mathbf{c}$ and then form
    $$\mathbf{x}^{(k)} = \sum_{j=0}^{k} \alpha_{jk}\mathbf{y}^{(j)}$$
  - *Krylov subspace methods*: find
    $$\mathbf{x}^{(k)} \in \operatorname{span}\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^k\mathbf{b}\}$$
    in a way that approximates the solution, i.e., $\mathbf{x}^{(k)} \approx \mathbf{x}$, in some sense
- splitting methods and semi-iterative methods are often called *stationary methods* to distinguish them from Krylov subspace methods (although this is not so clear cut — for example, conjugate gradient method, the oldest Krylov subspace method, may also be viewed as a semi-iterative method)

5. Splitting methods

- we want to solve $A\mathbf{x} = \mathbf{b}$ for $A \in \mathbb{R}^{n\times n}$ nonsingular
- we pick a suitable *splitting*
  $$A = M - N$$
  where M is nonsingular and easy to invert (not explicitly but in the sense that it is easy to solve $M\mathbf{x} = \mathbf{b}$ for any $\mathbf{b}$)
- from $A\mathbf{x} = \mathbf{b}$, we get
  $$M\mathbf{x} = N\mathbf{x} + \mathbf{b} \tag{5.1}$$
- this inspires the iteration
  $$M\mathbf{x}^{(k+1)} = N\mathbf{x}^{(k)} + \mathbf{b} \tag{5.2}$$
- subtracting (5.2) from (5.1), we obtain
  $$M(\mathbf{x} - \mathbf{x}^{(k+1)}) = N(\mathbf{x} - \mathbf{x}^{(k)})$$
- if we denote the *error* in $\mathbf{x}^{(k)}$ by $\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$, then
  $$\mathbf{e}^{(k+1)} = M^{-1}N\mathbf{e}^{(k)} =: B\mathbf{e}^{(k)}$$
- thus $\mathbf{e}^{(k)} = B\mathbf{e}^{(k)} = B^{k+1}\mathbf{e}^{(0)}$
- note that
  $$\mathbf{x}^{(k)} \to \mathbf{x} \quad \text{if and only if} \quad \mathbf{e}^{(k)} \to \mathbf{0} \quad \text{if and only if} \quad \|\mathbf{e}^{(k)}\| \to 0$$
- the matrix $B = M^{-1}N$ is somtimes called the *iteration matrix*
- its spectral radius $\rho(B)$ governs convergence rate, i.e., how quickly the error goes to zero
- recall that if $\rho(B^k) < 1$ then $\mathbf{e}^{(k)} \to \mathbf{0}$ for all choices of $\mathbf{x}^{(0)}$
- we have the following theorem:

Theorem 1. *$\mathbf{e}^{(k)} \to \mathbf{0}$ as $k \to \infty$ for all $\mathbf{e}^{(0)}$ if and only if $\rho(B) < 1$.*

Proof. Note that $\mathbf{e}^{(k)} = B^{k+1}\mathbf{e}^{(0)} \to \mathbf{0}$ for all $\mathbf{e}^{(0)}$ is equivalent to $\lim_{k\to\infty} B^k = O$ (the zero matrix) since we could choose $\mathbf{e}^{(0)}$ to be each of the standard basis vectors $\mathbf{e}_1, \dots, \mathbf{e}_n$ in turn and so we get
$$B^k = B^kI = B^k[\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] = [B^k\mathbf{e}_1, B^k\mathbf{e}_2, \dots, B^k\mathbf{e}_n] \to [\mathbf{0}, \mathbf{0}, \dots, \mathbf{0}] = O$$

as $k \to \infty$. Now by what we discussed in an earlier lecture (about the Jordan form), for a Jordan block,

$$J_r^k = \begin{bmatrix} \lambda_r^k & \binom{k}{1}\lambda_r^{k-1} & \binom{k}{2}\lambda_r^{k-2} & \cdots & \binom{k}{n_r-1}\lambda_r^{k-(n_r-1)} \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & & \lambda_r^k \end{bmatrix} \to O$$

as $k \to \infty$. Since B has a Jordan decomposition,

$$B = X \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_m \end{bmatrix} X^{-1},$$

we have

$$B^k = X \begin{bmatrix} J_1^k & & \\ & \ddots & \\ & & J_m^k \end{bmatrix} X^{-1} \to X \begin{bmatrix} O & & \\ & \ddots & \\ & & O \end{bmatrix} X^{-1} = O$$

as $k \to \infty$. □

- convergence can still occur if $\rho(B) = 1$, but in that case we must be careful in how we choose $\mathbf{x}^{(0)}$
- recall also that for all consistent norms,
$$\rho(B) \le \|B\|$$
and
$$\|B^k\| \le \|B\|^k$$
- from $\mathbf{e}^{(k)} = B^k \mathbf{e}^{(0)}$, it follows that
$$\frac{\|\mathbf{e}^{(k)}\|}{\|\mathbf{e}^{(0)}\|} \le \|B\|^k$$
- so if we find a consistent norm with $\|B\| < 1$, then this gives a sufficient condition for convergence
- note that convergence does not depend on the choice of norms since on finite-dimensional spaces, all norms are equivalent
- if we can prove statements like $\|B^k\| \to 0$ or $\|\mathbf{e}^{(k)}\| \to 0$ for any one norm, we know that it will hold for all norms

6. CONVERGENCE RATE

- formally, for a sequence $\mathbf{x}_k$ that converges to $\mathbf{x}$, its *convergence rate* $r \in (0, 1)$ is defined to be
$$r = \limsup_{k\to\infty} \frac{\|\mathbf{e}^{(k+1)}\|}{\|\mathbf{e}^{(k)}\|} = \limsup_{k\to\infty} \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}\|}{\|\mathbf{x}^{(k)} - \mathbf{x}\|}$$
or alternatively, the smallest $r \in (0, 1)$ such that
$$\|\mathbf{e}^{(k+1)}\| \le r\|\mathbf{e}^{(k)}\| \qquad \text{for all } k \text{ sufficiently large}$$
- a sequence that has such a property is called *linearly convergent* and we will often say that an iterative algorithm is linearly convergent for a class of problem if it generates a linearly convergent sequence for all choices of initial points $\mathbf{x}^{(0)}$

- if
$$\limsup_{k\to\infty} \frac{\|\mathbf{e}^{(k+1)}\|}{\|\mathbf{e}^{(k)}\|} = 0,$$
we say that the sequence (resp. algorithm) is *superlinearly convergent*
- if there exists $M > 0$ such that
$$\|\mathbf{e}^{(k+1)}\| \leq M\|\mathbf{e}^{(k)}\|^2 \qquad \text{for all } k \text{ sufficiently large,}$$
we say that the sequence (resp. algorithm) is *quadratically convergent*
- note that M does not need to be in $(0, 1)$
- more generally the largest p for which there exists $M > 0$ such that
$$\|\mathbf{e}^{(k+1)}\| \leq M\|\mathbf{e}^{(k)}\|^p \qquad \text{for all } k \text{ sufficiently large,}$$
is called the *order of convergence*