Welcome to Ma221 - Fall 24 -

Lecture 1

people.eecs.berkeley.edu/~demmel/ma221_Fall24

see webpage for

- class notes
  - last semester's notes
  - (latex version on bcourses
- office hours
- Grader - Sadhana Kulkarni
- Grading - weekly homework (#1 posted)
  - group projects
  - no exams
- class Survey

Notation

$\|x\|_2 = \sqrt{\sum_i |x_i|^2}$ - 2 norm

$\operatorname{argmin}_x f(x)$ = value of x that minimizes $f(x)$

$f(n) = O(g(n))$ means $|f(n)| \le C\,|g(n)|$ for some $C>0$ and n large enough

$f(n) = \Omega(g(n))$ means
$|f(n)| \geq C|g(n)|$ for $C > 0$, n large enough

$f(n) = \Theta(g(n))$ means $f(n) = O(g(n))$
and $f(n) = \Omega(g(n))$

Syllabus – 4 "axes" of design space of linear algebra problems

1) mathematical problem
   - solve $Ax = b$
   - least squares $\arg\min_x \|Ax - b\|_2$
   - eigenproblems $Ax = \lambda x$
   - many generalizations

2) structure of A
   - dense, symmetric ($A = A^T$)
     - positive definite
   - sparse, lots of zeros
   - "structured" eg
     - Toeplitz: $A(i,j) = x_{i-j}$
       - A constant along diagonals

3) desired accuracy: spectrum including
   - guaranteed correct (too expensive)
   - "guaranteed correct" except for "rare cases"

ex: iterative refinement for $Ax=b$ using mixed precision

"backward stable" - exact answer for slightly wrong problem
"gold standard"

residual as small as desired

"probably OK" (randomized algorithms)

error bounds

4) as fast as possible on your target computer:
your laptop (might have GPU...)
big parallel computer,
cloud, cell-phone, ...

"problem" = choice from ~~axes~~ 1), 2), 3), 4)
Answer could be
"type A\b"

"download standard SW from this URL"

"project available to implement proposed algorithm"

"open problem"

All combinations of problems from 1), ..., 4) would take >>1 semester, so we will present a subset, may depend on class survey.

Axis 1): Math problem

Solve $Ax=b$: well defined for A square full rank, otherwise (or A close to matrix with lower rank) then may want least squares

Least squares:

Overdetermined: $\underset{x}{\operatorname{argmin}} \|Ax-b\|_2$

when $A^{m\times n}$ $m \geq n$ and full column rank

A not full rank $\Rightarrow x$ not unique

so can pick x that also minimizes $\|x\|_2$ to make it unique

Ridge Regression:

$\underset{x}{\operatorname{argmin}} \|Ax-b\|_2^2 + \lambda \|x\|_2^2$

for $\lambda > 0$ (also called Tikhonov regularization)

solution unique if $\lambda > 0$

Constrained LS: $\underset{x:Cx=d}{\operatorname{argmin}} \|Ax-b\|_2$

Ex: x = fractions of a population

$\Rightarrow \sum_i x_i = 1$

(seems natural to ask $x_i \geq 0$, harder)

Weighted LS $\quad \underset{x}{\operatorname{argmin}} \| W^{-1}(Ax-b)\|_2$

where W = weight matrix, full rank

(Gauss-Markov linear model)

Total Least Squares

$$\underset{x:\ (A+E)x = b+r}{\operatorname{argmin}} \| [E, r] \|_2$$

Eigen problems:

Notation: $Ax_i = \lambda_i x_i \quad x_i \neq 0$

for $i = 1, \ldots, n$, $X^{n \times n} = [x_1, \ldots, x_n]$

$\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$

$AX = X\Lambda$, if X invertible:

eigen decomposition $A = X \Lambda X^{-1}$

Recall A may not have n independent evecs

eg $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Earlier LA class: Jordan Form:

but discontinuous $A' = \begin{bmatrix} \varepsilon & 1 \\ 0 & 0 \end{bmatrix}$: has Jordan form $\begin{bmatrix} \varepsilon & 0 \\ 0 & 0 \end{bmatrix}$ $\forall \varepsilon \neq 0$

$\Rightarrow$ numerically unstable

We will compute Schur form instead

SVD: Singular Value Decomposition

$$A^{m\times n} = U\Sigma V^T \qquad m \geq n$$

$U^{m\times m}$ orthogonal: $UU^T = I$

$V^{n\times n}$ orthogonal

$\Sigma^{m\times n}$ and diagonal
$$\begin{array}{cc} & n \\ m & \begin{bmatrix} \ddots & 0 \\ & \\ 0 & \end{bmatrix} \end{array}$$

with diagonal entries

$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n$

called singular values

columns of U and V are

Left and right singular vectors:

$$\begin{aligned}
A \cdot A^T &= (U\Sigma V^T)(U\Sigma V^T)^T \\
&= U\Sigma V^T (V^T)^T \Sigma^T U^T \\
&= U\Sigma \underbrace{V^T V}_{I} \Sigma^T U^T \\
&= U \underbrace{\Sigma\Sigma^T}_{\text{diagonal}} U^T
\end{aligned}$$

$=$ eigendecomposition of AA^T

$$\begin{aligned}
A^T A &= (U\Sigma V^T)^T (U\Sigma V^T) \\
&= V\Sigma^T \underbrace{U^T U}_{I} \Sigma V^T \\
&= V\Sigma^T\Sigma V^T
\end{aligned}$$

$=$ eigendecomp of $A^T A$

also SVD solves LS problem

Invariant subspaces: $x'(t) = A\,x(t)$

$x(0)$ given: Suppose $A\,x(0) = \lambda\,x(0)$

then $x(t) = e^{\lambda t} x(0)$

Easy to tell if $x(t) \to 0$ as $t \to \infty$

depends on whether $\text{Real}(\lambda) < 0$

$x(0) = \sum_{i=1}^{n} \beta_i x_i$ where $A x_i = \lambda_i x_i$

$\Rightarrow x(t) = \sum_i e^{\lambda_i t} \beta_i x_i$

whether $x(t) \to 0$ as $t \to \infty$ depends on whether all $\text{Real}(\lambda_i) < 0$ when $\beta_i \neq 0$

The space spanned by all x_i with $\text{real}(\lambda_i) < 0$ is called invariant space V:

$\forall\, x \in V \quad Ax \in V$

Often possible to have algorithm for an invariant subspace that is faster, more accurate than individual evecs

Generalized Eigenproblems:

Consider $M x''(t) + K \cdot x(t) = 0$

Ex: $x(t)$: positions

M: mass , K: stiffness

$x(t)$: currents in circuit

M: inductances

K: reciprocals of capacitances

plug in $x(t) = e^{\lambda t} \cdot x(0) \Rightarrow$

$\lambda^2 M x(0) + K x(0) = 0$

$\Rightarrow x(0)$ is a generalized evec

λ^2 is a " eval

of (M, K)

usual def of eval: $\det(K - \lambda I) = 0$

becomes $\det(K + \lambda' M) = 0$ where $\lambda' = \lambda^2$

all ideas and algs for one matrix generalize:

Jordan form $\rightarrow$ Weierstrass form

Schur $\rightarrow$ generalized Schur

Why not solve for evals of $M^{-1}K$?

M could be singular, or nearly so:

Singular M arises in "differential algebraic equations" aka ODE with linear constraints

Nonlinear eigenproblems:

$$M x''(t) + D\cdot x'(t) + K\cdot x(t) = 0$$

D : damping matrix (friction) x positions

: resistances x currents

plug in $x(t) = e^{\lambda t} x(0)$, get

$$\lambda^2 M x(0) + \lambda D x(0) + K x(0) = 0$$

can reduce to linear problem of twice size

Singular eigenproblems: Control Systems

$$x'(t) = A\, x(t) + B u(t)$$

$A^{n \times n}$ $B^{n \times m}$ $m < n$

$u(t)$ = control input to be chosen to "control" $x(t)$

What subspace can $x(t)$ lie in and be controlled by $u(t)$?

Can be formulated as a rectangular eigenproblem for $n\overset{m+n}{[B, A]}$, $n\overset{m+n}{[0, I]}$

Jordan form becomes Kronecker form

Schur form generalizes too

Partial Solution:

Ex: subset of evals, evecs instead of all

Ex: low rank approximation of your big input matrix

— cheaper, sometimes more accurate

Updating Solutions

given existing solution for A and A changes "a little", compute new answer "cheaply"

"a little" could mean

change a few entries

a few rows and columns

add a low rank matrix to A

Eg: SVD alg depends on this

Tensors (not) 3D, 4D etc arrays instead of 2D arrays (matrices)

lots of analogous problems:

matrix multiply, low rank approx

Tensors often harder, use matrix algs as building blocks

"Most tensor problems are NP complete"

Axis 2: structure of A.

Office hours story:

Student: "I need to solve an $n \times n$ linear system $Ax=b$. What should I do?"

Prof: Standard Alg is Gauss Elim (GE) costs $\frac{2}{3}n^3$ floating point ops (flops)

S: Too expensive

P: Tell me more about A

S: Well, A real symmetric $A=A^T$

P: Anything else?

S: Oh yes, A pos definite, $x^TAx>0 \ \forall x \neq 0$

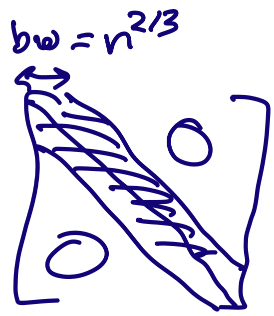
P: Great: You can use Cholesky, only costs $\frac{1}{3}n^3$ flops, half of GE

S: Still too expensive

Prof. recording conversation on board as decision tree, node for each alg, edge for each question/answer (see Table 6.1 in text)

P: Tell me more

S: A has lots of zeros, in fact zero if farther than $n^{2/3}$ away from diagonal



P: Great, you have a band matrix
bw = bandwidth = $n^{2/3}$
Version of Cholesky

$$\text{cost} = O(bw^2 \cdot n) = O(n^{7/3}) \text{ flops} -$$

much cheaper!

S: Still too expensive

P: Tell me more:

S: I need to solve $Ax=b$ many times for different b. Should I compute A^{-1} and reuse it?

P: A^{-1} will be dense, so $A^{-1} \cdot b$ costs $2n^2$ flops, but can reuse output of band Cholesky to solve for each b in $O(bw \cdot n) = O(n^{5/3})$ flops

S: Still too expensive

P: Tell me more

S: There are actually a lot more zeroes

at most 7 per row

P: Let's try an iterative method instead of direct. Only need to do $A \cdot x$ over and over, updating an approximate solution, $A \cdot x$ costs $O(n)$ flops

S: How many $A \cdot x$ do I need?

P: can you tell me about range of evals of A? eg condition number $= \lambda_{max} / \lambda_{min}$

S: Yes $\lambda_{max} / \lambda_{min} = n^{2/3}$ too

P: You can use Conjugate Gradients (CG), needs $O\left(\sqrt{\frac{\lambda_{max}}{\lambda_{min}}}\right) = O(n^{1/3})$ $A \cdot x$ operations, $\text{cost} = O(n \cdot n^{1/3}) = O(n^{4/3})$ Happy yet?

S: No

P: tell me more

S: I know λ_{max}, λ_{min}, does that help?

P: What are you really trying to solve?

S: I have a cube of metal
I know temp on surface
I want temp everywhere inside

P: Oh, you're solving 3D Poisson equation, Best could be either
direct: FFT = Fast Fourier Transform
cost $O(n \log n)$
Iterative Multigrid (MG)
cost $O(n)$ flops
$\Rightarrow O(1)$ per component of x
$\Rightarrow$ lower bound

S: So where can I get software?

Important to exploit structure.
Poisson Eq best studied structure
many others