Welcome to Ma 221 !

Numerical Linear Algebra Spr22

people.eecs.berkeley.edu/~demmel/ma221_Spr22

see webpage for

- class notes
  - last semester's notes
  - latex version on bcourses
- office hours
- GSI - Michael Heinz
- Grading - weekly homework (#1 posted)
  - group projects
    - submit preproposal
  - no exams
- Class survey

Notation

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

$\arg\min_x f(x)$ = value of x that minimizes $f(x)$

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

$f(n) = \Omega(g(n))$ means
$|f(n)| \ge C|g(n)|$ for $C>0$, n large enough
$f(n) = \Theta(g(n))$ if $f=O(g)$ and $f=\Omega(g)$

Syllabus - 4 "axes" of design
space of linear algebra algorithms

1) mathematical problem
   - solve $Ax=b$
   - least squares $\underset{x}{\operatorname{argmin}} \|Ax-b\|_2$
   - eigenproblems $Ax=\lambda x$
   - many generalizations...

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

3) desired accuracy: spectrum
guaranteed correct
"guaranteed correct" except for
"rare cases"
"backward stable"
residual as small as desired
"probably OK" (randomized algorithms)

---

error bounds

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

"problem" = choice from 1), 2), 3), 4)
Answer could be

"type A\b"
"download standard SW from this URL"
"project available to implement a
proposed algorithm"
"open problem"

All combinations of problems from 1) --- 4) would take >> 1 semester, so we will explore important subset. Could be tweaked depending on class survey

---

Axis 1): Math problem

Solve $Ax=b$: well defined for A square, full rank, otherwise (or A close to matrix of low rank), then least squares may be better

Least Squares

Overdetermined:

$\underset{x}{\text{argmin}} \|Ax-b\|_2$ when

$A^{m \times n}$ has full column rank ($m \geq n$)

A not full rank $\Rightarrow$ x not unique

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

Ridge Regression:

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

for $\lambda \geq 0$ (also called Tikhonov regularization). Solution unique if $\lambda > 0$

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

Ex: x: fractions of population
$\Rightarrow \sum_{i=1}^{n} x_i = 1$ (seems natural to ask $x_i \geq 0$, harder)

Weighted LS: $\underset{x}{\operatorname{argmin}} \|W^{-1}(Ax-b)\|_2$
where W 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 $i = 1..n \quad X^{n\times n} = [x_1, \dots, x_n]$

$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$

$AX = X\Lambda$, X invertible $\Rightarrow$

eigendecomposition $A = X\Lambda X^{-1}$

Recall A may not have n independent evecs, eg: $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Earlier LA class: Jordan Form

but this is numerically unstable

We will use 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 $\begin{array}{cc} & n \\ m & \begin{bmatrix} \ddots & 0 \\ 0 & \ddots \end{bmatrix} \end{array}$

with diagonal entries

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)}$

called singular values

Columns of U and V are
left and right singular vectors, resp.

$$A \cdot A^T = (U \Sigma V^T)(U \Sigma V^T)^T$$
$$= U \Sigma \underbrace{V^T V}_{I} \Sigma^T U^T$$
$$= U \underbrace{\Sigma \Sigma^T}_{\text{diagonal}} U^T$$
$$= \text{eigendecomposition of } AA^T$$

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T)$$
$$= V \underbrace{\Sigma^T \Sigma}_{\text{diagonal}} V^T$$
$$= \text{eigendecomposition of } A^T A$$

SVD most "reliable" method for LS,
also most expensive

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 $\text{Real}(\lambda) < 0$

$x(0) = \sum_i \beta_i x_i$ 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 $\mathrm{Real}(\lambda_i) < 0$ when $\beta_i \neq 0$

i.e. whether $x(0)$ in subspace spanned by all evecs x_i where $\mathrm{Real}(\lambda_i) < 0$

This called "invariant subspace"

Often possible to have algorithm to compute invariant subspace that is cheaper + more accurate than computing all evecs

Generalized Eigenproblems

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

Ex: $x(t)$ positions

M: Mass, K: Stiffness

$x(t)$: currents in circuit

M: inductances, K: reciprocals of capacitances plug $x(t) = e^{\lambda t} x(0) \Rightarrow$

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

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

λ^2 " " " 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 → Weierstrass form

Schur → generalized Schur

Note if M singular, can't convert to standard problem for $M^{-1}K$;

Singular M arises in

"differential algebraic equations"

aka ODEs with linear constraints

Nonlinear eigenproblems:

$$Mx''(t) + D\cdot x'(t) + Kx(t) = 0$$

D: damping matrix if x positions
resistances if x currents

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

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

We will reduce to linear problem of twice size

Singular eigenproblems: Control Systems:

$$x'(t) = Ax(t) + Bu(t)$$

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

$u(t)$ 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 rectangular eigenproblem for $\underset{n}{[\overset{m+n}{B, A}]}, \underset{n}{[\overset{m+n}{O, I}]}$)

(Jordan form becomes Kronecker form)

Partial Solution: Instead of all evals & evecs, compute subset

Ex: invariant subspace

low-rank approximations

- cheaper, more accurate

Updating Solution

if I've solved problem for A and A changes "a little", update solution cheaply:

little could mean: change a few entries, few rows or columns, add a few rows and columns, add a low rank matrix to A

Tensors (not) instead of 2D arrays (matrices) input is 3D, 4D, higher

Lots of problems extend:

matrix multiply, low rank approx

Tensors often harder, but use matrices as building blocks

Axis 2: Structure of A:

Story about office hours

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

Professor: "Standard Algorithm is Gaussian elimination (GE), costs $\frac{2}{3}n^3$ floating operations (flops)"

S: Too expensive

P: Tell me more about A

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

P: Anything else?

S: Oh yes, A is positive definite $x^T A x > 0$ for all nonzero x

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

Professor records conversation on board as a "decision tree", where each node = algorithm

and each edge = property of A
(see Table 6.1 in text)

S: Still too expensive

P: Tell me more

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

P: Great! band matrix

version of band Cholesky cost $O(bw^2 \cdot n) = O(n^{7/3})$ flops, much cheaper!

S: Still too expensive

P: Tell me more

S: I need to solve problem many times with same A, different b. So should I just compute A^{-1}, and multiply by it?

P: A^{-1} will be dense, so multiplying $A^{-1}b$ cost $2n^2$ flops, but can reuse the 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 many more zeros, just ≤ 7 nonzeros/row

P: Let's try an iterative algorithm instead of a direct method. Iterative method only needs to do $A \cdot x$ over and over, updating an approximate answer, and $A \cdot x$ costs $O(n)$ flops

S: How many matrix-vector multiplies do I need?

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

S: Yes $K(A) \approx n^{2/3}$ too

P: You could use Conjugate Gradients (CG) needs $O(\sqrt{\kappa(A)}) = O(n^{1/3})$ matrix-vector multiplies

$\Rightarrow$ 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: You know a lot about A. What problem are you really trying to solve?

S: I have a cube of metal, I know temp on faces of cube I want want temp inside

P: Oh you're solving 3D Poisson Eq!

Best choice of algorithm

is either

direct: FFT = Fast Fourier Transform

cost $O(n \log n)$ flops

iterative: Multigrid (MG)

cost $O(n)$ flops, $O(1)$

per component of x

$\Longrightarrow$ lower bound

S: So where can I download the software?

Important: exploit math. structure

Poisson Eq one of best studied

linear systems, but many more...