Cheat Sheet MAT261

Tommy Odland – Num. Lin. Alg. – Edited: November 30, 2016

▷ Fundamentals

- Vector norms are $\|x\|_p$, where p is usually in $\{1, 2, \infty\}$.
- Matrix norms include the Frobenius norm $\|A\|_F$ and the induced vector norms

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p$$

- The SVD (singular value decomposition) of A is $A = U\Sigma V^*$, where U and V are orthogonal and Σ is diagonal. There are full and reduced versions. For hermitian ($A^* = A$) matrices, $\text{eig}(A) = \text{svd}(A)$.

▷ QR and least SQ

- A projection matrix is idempotent $P^2 = P$. The orthogonal projection onto q is $P_q = \frac{qq^*}{q^*q}$. The complementary projector is $P_{\perp q} = I - \frac{qq^*}{q^*q}$.
- The **QR-factorization** of A is $A = QR$, where Q is orthogonal and R is upper triangular. It comes in a full and reduced flavor.
- **Gram-Schmidt** takes a set of vectors $\langle a_1, a_2, ..., a_m\rangle$ and creates an orthogonal basis for the same space $\langle q_1, q_2, ..., q_m\rangle$. Classical is unstable, it "looks back". Modified is stable, it "looks ahead". MGS has operation count $\sim 2mn^2$.
- **Householder** reflectors F are an involution $F^2 = I$. Householder computes $A = QR$ by orthogonal triangularization, while Gram-Schmidt is a process of triangular orthogonalization. QR by Householder has operation count $\sim 2mn^2 - \frac{2}{3}n^3$.
- The least SQ problem is to minimize $\|r\|_2 = \|b - Ax\|_2$. The normal equations are $A^*Ax = A^*b$. Least SQ is solved by Cholesky on normal eqns (fast), QR-factorization (standard method, good, stable) or by the SVD (good when A is close to rank-deficient).

▷ Conditioning, stability

- Problems $f : X \to Y$ have **conditioning**. X is the input data, and Y is the solution. Conditioning measures how sensitive the problem is to a perturbation of the input.
- The relative condition number is

$$\kappa = \lim_{\delta\to 0} \sup_{\|\delta x\|<\delta} \left(\frac{\|\delta f\|}{\|f(x)\|} / \frac{\|\delta x\|}{\|x\|} \right)$$

- Conditioning of a matrix $\kappa(A) = \|A\|\|A^{-1}\|$. In the 2 norm, $\|A\|\|A^{-1}\| = \sigma_1/\sigma_m$.
- The conditioning of a system of equations $f : A \to x$ is $\kappa(A)$. This is the same conditioning as $f : A \to b$, i.e. matrix-vector multiplication.
- The smallest number representable on a computer is $\epsilon_{\text{machine}}$, henceforth denoted ϵ_m. On my system, it's $\approx 2 \times 10^{-16}$. It's the smallest gap in floating arithmetic representation, a subset of the reals $\mathbb{F} \subset \mathbb{R}$.
- The fundamental axiom of floating point arithmetic is: There exists $|\epsilon| \leq \epsilon_m$ so that

$$(x \circledast y) = (x * y)(1 + \epsilon)$$

- A problem is modeled as $f : X \to Y$, and an algorithm is modeled as $\tilde{f} : X \to Y$. An algorithm may be accurate, stable or backward stable. *Accurate* if the computed solution is close to the true solution. *Stable* if a small perturbation of the input results in small perturbation of the output (it does not grow unbounded). *Backward stable* if the algorithm solves a perturbed problem exactly.
- To prove an algorithm $\tilde{f} : X \to Y$ backward stable, one must express every floating point error as a perturbation of the input X.
- The accuracy of a backward stable algorithm is given by

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O\left(\kappa(x)\ \epsilon_m\right)$$

- Most numerical algorithms in use are backward stable. QR by Householder is backward stable, however $\tilde{Q}$ and $\tilde{R}$ are individually no good. Errors are "diabolically correlated."

- Theorems on the conditioning of the least SQ problem ($\min_x \|b - Ax\|_2$) exist, both in terms of $f : b \to x$ and $f : A \to x$.
- The least SQ problem may be solved by QR with Householder (good, backward stable), MATLAB's `x = A \b;` (very good, QR with column pivoting). The normal equations $A^*Ax = A^*b$ should never be used, because $\kappa(A)$ is squared when forming A^*A.

▷ System of equations

- The **LU-factorization** of A is $A = LU$, where U is upper-triangular and L is lower triangular with 1's on the diagonal. The operation count of LU-factorization by Gaussian elimination is $\sim \frac{2}{3}m^3$. Without pivoting, Gaussian elimination is neither stable nor backward stable.
- Complete pivoting searches for maximum pivot in rows and columns. Partial pivoting searches in rows only, forming $PA = LU$. Partial achieves good results with less work. Pivoting controls instability, in L, all subdiagonal entries become < 1.
- The growth factor is defined as

$$\rho = \frac{\max_{i,j}|u_{ij}|}{\max_{i,j}|a_{ij}|}$$

  If ρ is of order 1, elimination is stable. In general $\|\delta A\|/\|A\| = O(\rho\epsilon_m)$. The worst case is $\rho = 2^{m-1}$. Useless in practice, but also extremely rare in practice.
- **Cholesky factorization** is LU-factorization of a symmetric matrix, so that $A = R^*R$, where R is upper-triangular. Operation count is half of Gaussian elimination, i.e. $\sim m^3/3$. Always stable, no need for pivoting.

▷ Eigenvalues

- The eigenvalue decomposition of a matrix is $A = X\Lambda X^{-1}$. The characteristic polynomial is $p_A(z) = \det(A - Iz)$, where $p_A(\lambda) = 0 \Leftrightarrow \lambda$ is an eigenvalue.
- The **Schur** factorization of A is $A = QTQ^*$, where T is upper-triangular. It's an eigenvalue-revealing factorization. When $A = A^*$, T is diagonal.
- The **Hessenberg** factorization is $A = QHQ^*$, where H is in Hessenberg form (all entries below sub-diagonal are 0). When $A = A^*$, H is tridiagonal.
- Eigenvalue algorithms usually consist of the following steps:
  - Direct: Bring A to Hessenberg form in $\sim \frac{4}{3}m^3$ operations (Hermitian A)
  - Iterative: Bring A to tridiagonal (diagonal if $A^* = A$) in $O(m^2)$ flops
- Householder transforms A into Hessenberg form H using $\sim \frac{10}{3}m^3$ operations. If $A = A^*$, the operation count is reduced to $\sim \frac{4}{3}m^3$.
- **Power iteration** is (1) choose random v, (2) multiply by A and normalize, (3) repeat (2). It will yield the largest eigenvector. Let $|\lambda_1| > |\lambda_2| > ...$, then convergence is $O(|\frac{\lambda_2}{\lambda_1}|^k)$.
- **Inverse iteration** is based on the fact that A and $(A-\mu I)^{-1}$ have the same eigenvalue. We use power iteration on $(A-\mu I)^{-1}$, where μ is an eigenvalue estimate.
- The **Rayleigh quotient** is the solution to $\min_\alpha \|Ax - \alpha x\|$, and it's given by

$$r(x) = \frac{x^TAx}{x^Tx}$$

  When x is an eigenvector v, $r(v) = \lambda$.
- Rayleigh quotient iteration combines inverse iteration (finds eigenvector) with the Rayleigh quotient (find eigenvalue) to achieve cubic convergence $O(\epsilon^3)$.
- The **QR-algorithm** takes the QR-factorization of A, multiplies back in reverse order, and repeats. It converges to a Schur-factorization of A. It's a stable approach to the more intuitive simultaneous iteration. Using shifts speeds up convergence.

	quasi-direct	**iterative**
unstable	simul. iter.	QR of K_n
subtle, stable	QR-algorithm	Arnoldi

- Shifted QR-algorithm is backward stable, cost is $\sim \frac{4}{3}m^3$, with cubic convergence.
- The **Jacobi-algorithm** for symmetric A is based on orthogonal rotations. It "rotates" every off-diagonal entry in sweeps. Every sweep reduces the size of off-diagonals. Quadratic convergence. Based on similarity transform J^TAJ.

- The **bisection** algorithm works on tridiagonal A. It's used to find sub-sections of eigenvalues. Based on sign changes in $\det\left(A^{(1)}\right), \det\left(A^{(2)}\right), \det\left(A^{(3)}\right), ...,$ the eigenvalue interlace property and the determinant three-term recurrence.
- Computing the SVD in two phases: (1) bidiagonalize with Householder then (2) chase zeros. Bidiagonalization is done in $\sim 4mn^2 - \frac{4}{3}n^3$ flops. If $m \gg n$, use QR-factorization first.

▷ Iterative methods

	$Ax = b$	$Ax = \lambda x$
$A = A^*$	CG	Lanczos
$A \neq A^*$	GMRES, CGN, BCG	Arnoldi

- Krylov subspace $\mathcal{K}_n = \langle b, Ab, A^2b, ..., A^{n-1}b \rangle$
- Krylov matrix $K_n = [b|Ab|A^2b|...|A^{n-1}b]$
- Krylov methods allows one to black box matrix multiplication , so that $A : x \to Ax$.
- The Arnoldi iteration starts with $AQ_n = Q_{n+1}\tilde{H}_n$. We run modified Gram-Schmidt on $Aq_n = h_{1n}q_1 + ... + h_{nn}q_n$. For every n, $\langle q_1, ..., q_n \rangle$ is an orthonormal basis for $\mathcal{K}_n$.
- The eigenvalues of $\tilde{H}_n$ in $A = QHQ^*$ are the Ritz values – approximations to the eigenvalues of A. Extreme eigenvalues are found first. The polynomial approximation problem is $\|p^n(A)b\|$ = minimum, where $p^n \in P^n = \{\text{polynomials with } 1x^n + ...\}$.
- **GMRES** (generalized minimal residuals) solves $Ax = b$. At each step, GMRES finds $x_n \in \mathcal{K}_n$ such that the norm of the residual $\|r_n\| = \|b - Ax_n\|$ is minimized. The algorithm uses Arnoldi iterations. Convergence is monotonic, the residual at step m is 0, and convergence depends on eigenvalues.
- The **Lanczos iteration** is like Arnoldi for symmetric matrices, the equation becomes $AQ_n = Q_{n+1}\tilde{T}_n$, where T is tridiagonal.
- The **CG** (Conjugate Gradient) algorithm minimizes $\phi(x) = \frac{1}{2}x^TAx - x^Tb$ by generating $x_n \in \mathcal{K}_n$ such that the error $\|e_n\|_A$ is minimized at each step.
  - Properties of CG include orthogonal residuals $r_i^T f_j = 0$, A-conjugate ($p_i^T A p_j = 0$) search directions p, x_n is the unique minimizer in $\mathcal{K}_n$ and monotonic convergence. Top algorithm for solving $Ax = b$ when $A = A^* > 0$ (s.p.d).
  - Two important convergence theorems are

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)|$$

    and

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^n$$

    For small $\kappa(A)$, convergence is superlinear. Preconditioning can speed up convergence.
- CGN (Conjugate Gradients on Normal equations) is a method where $Ax = b$ is solved ($A \neq A^T$) by solving $A^*Ax = A^*b$ by CG. The condition number is squared, and convergence is determined by singular values of A, not the eigenvalues as with CG.
- Tridiagonal biorthogonalization (or nonsymmetric Lanczos iteration) takes a nonsymmetric A to $A = VTV^{-1}$ iteratively. T is tridiagonal, but V is not orthogonal.
- Biconjugate gradients (BCG) solves $Ax = b$ for non-symmetric A. Like CG, but with 2 search directions Other variants include QMR (quasi-minimal residuals), Bi-CGSTAB is version of BCG with smoother convergence.
- Preconditioning aims to solve $M^{-1}Ax = M^{-1}b$ instead of $Ax = b$. M^{-1} should be close to A^{-1}. Trivial cases are $M^{-1} = A^{-1}$ and $M^{-1} = I$. To preserve a hermitian A, we can set $M = CC^*$. $M^{-1}A$ should be close to normal. Some preconditioners are $M = \text{diag}(A)$, $M = \text{triu}(A)$ and incomplete Cholesky. Preconditioners are very important in practical applications.

▷ References

- Lloyd N. Trefethen. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997.