CS 598 EVS: Tensor Computations

Matrix Computations Background

Edgar Solomonik

University of Illinois at Urbana-Champaign

Conditioning

- **Absolute Condition Number**:

- **(Relative) Condition Number**:

Posedness and Conditioning

- **What is the condition number of an ill-posed problem?**

Matrix Condition Number

- The matrix condition number $\kappa(\boldsymbol{A})$ is the ratio between the max and min distance from the surface to the center of the unit ball (norm-1 vectors) transformed by $\boldsymbol{A}$:

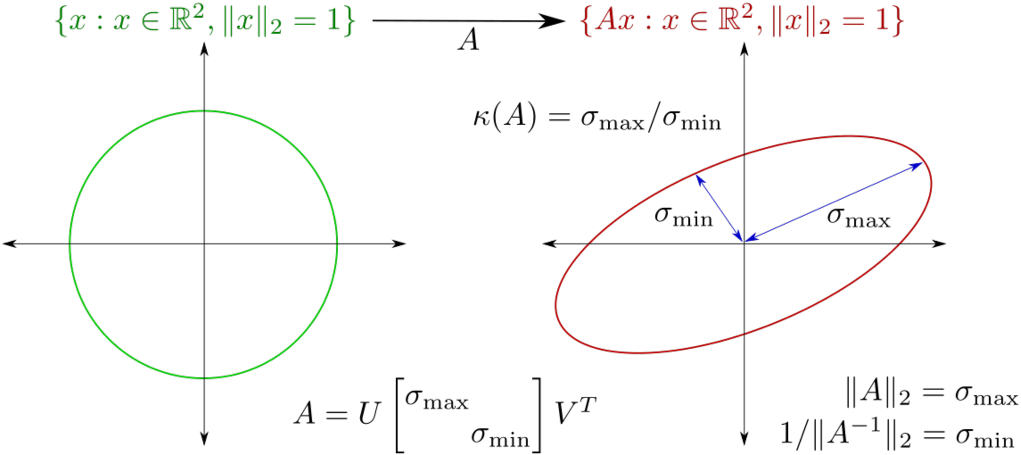
- The matrix condition number bounds the worst-case amplification of error in a matrix-vector product:

Singular Value Decomposition

- The singular value decomposition (SVD)

- Condition number in terms of singular values

Visualization of Matrix Conditioning



$\{x : x \in \mathbb{R}^2, \|x\|_2 = 1\} \xrightarrow[A]{} \{Ax : x \in \mathbb{R}^2, \|x\|_2 = 1\}$

$\kappa(A) = \sigma_{\max}/\sigma_{\min}$

$\sigma_{\min}$

$\sigma_{\max}$

$$A = U \begin{bmatrix} \sigma_{\max} & \\ & \sigma_{\min} \end{bmatrix} V^T$$

$$\|A\|_2 = \sigma_{\max}$$

$$1/\|A^{-1}\|_2 = \sigma_{\min}$$

Linear Least Squares

- Find $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$:

- Given the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ we have $\boldsymbol{x}^\star = \underbrace{\boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T}_{\boldsymbol{A}^\dagger}\boldsymbol{b}$, where $\boldsymbol{\Sigma}^\dagger$ contains the reciprocal of all nonzeros in $\boldsymbol{\Sigma}$, and more generally † denotes pseudoinverse:

Demo: *Normal equations vs Pseudoinverse*
Demo: *Issues with the normal equations*

Normal Equations

- *Normal equations* are given by solving $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$:
- However, solving the normal equations is a more ill-conditioned problem then the original least squares algorithm

Solving the Normal Equations

- If $\boldsymbol{A}$ is full-rank, then $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric positive definite (SPD):
- Since $\boldsymbol{A}^T\boldsymbol{A}$ is SPD we can use Cholesky factorization, to factorize it and solve linear systems:

QR Factorization

- If $\boldsymbol{A}$ is full-rank there exists an orthogonal matrix $\boldsymbol{Q}$ and a unique upper-triangular matrix $\boldsymbol{R}$ with a positive diagonal such that $\boldsymbol{A} = \boldsymbol{QR}$

- A reduced QR factorization (unique part of general QR) is defined so that $\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ has orthonormal columns and $\boldsymbol{R}$ is square and upper-triangular

- We can solve the normal equations (and consequently the linear least squares problem) via reduced QR as follows

Computing the QR Factorization

- The Cholesky-QR algorithm uses the normal equations to obtain the QR factorization

- Orthogonalization-based methods are most efficient and stable for QR factorization of dense matrices

Eigenvalue Decomposition

- If a matrix $\boldsymbol{A}$ is diagonalizable, it has an *eigenvalue decomposition*

- $\boldsymbol{A}$ and $\boldsymbol{B}$ are *similar*, if there exist $\boldsymbol{Z}$ such that $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^{-1}$

Similarity of Matrices

matrix	*similarity*	*reduced form*
SPD		
real symmetric		
Hermitian		
normal		
real		
diagonalizable		
arbitrary		

Rayleigh Quotient

- For any vector $\boldsymbol{x}$ that is close to an eigenvector, the *Rayleigh quotient* provides an estimate of the associated eigenvalue of $\boldsymbol{A}$:

Introduction to Krylov Subspace Methods

- *Krylov subspace methods* work with information contained in the $n \times k$ matrix

$$\boldsymbol{K}_k = \begin{bmatrix} \boldsymbol{x_0} & \boldsymbol{A}\boldsymbol{x_0} & \cdots & \boldsymbol{A}^{k-1}\boldsymbol{x_0} \end{bmatrix}$$

- $\boldsymbol{A}$ is similar to *companion matrix* $\boldsymbol{C} = \boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{K}_n$:

Krylov Subspaces

- Given $\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{K}_k$, we obtain an orthonormal basis for the Krylov subspace,

  $$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{x}_0) = span(\boldsymbol{Q}_k) = \{p(\boldsymbol{A})\boldsymbol{x}_0 : deg(p) < k\},$$

  where p is any polynomial of degree less than k.
- The Krylov subspace includes the $k-1$ approximate dominant eigenvectors generated by $k-1$ steps of power iteration:

Krylov Subspace Methods

- The $k \times k$ matrix $\boldsymbol{H}_k = \boldsymbol{Q}_k^T \boldsymbol{A} \boldsymbol{Q}_k$ minimizes $||\boldsymbol{A}\boldsymbol{Q}_k - \boldsymbol{Q}_k \boldsymbol{H}_k||_2$:
- $\boldsymbol{H}_k$ is upper-Hessenberg, because the companion matrix $\boldsymbol{C}_n$ is upper-Hessenberg:

Rayleigh-Ritz Procedure

- The eigenvalues/eigenvectors of $\boldsymbol{H}_k$ are the *Ritz values/vectors*:

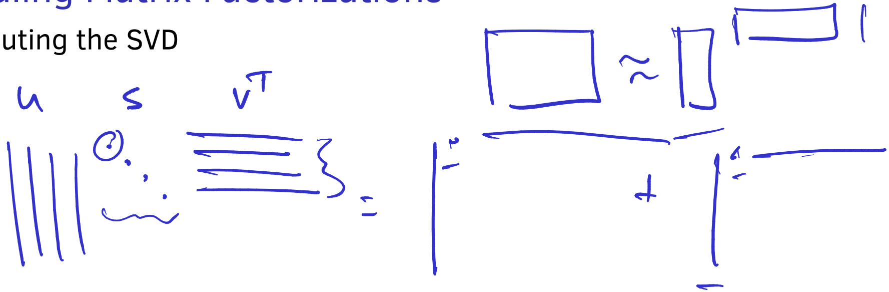
- The Ritz vectors and values are the *ideal approximations* of the actual eigenvalues and eigenvectors based on only $\boldsymbol{H}_k$ and $\boldsymbol{Q}_k$:

Low Rank Matrix Approximation

- Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ seek rank $r < m, n$ approximation
- Eckart-Young (optimal low-rank approximation by SVD) theorem

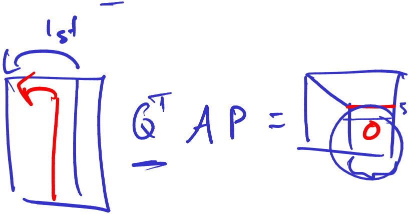
Rank Revealing Matrix Factorizations

- Computing the SVD



- QR with column pivoting

$$AP = QR$$



Orthogonal Iteration

- For sparse matrices, QR factorization creates fill, so must revert to iterative methods

$$Q_{k+1} \leftarrow QR\Big(\underbrace{\overset{\text{symmetrize}}{\overset{\downarrow}{A}}}_{\underline{A^TA}}Q_k\Big)$$

- Orthogonal iteration interleaves deflation and power iteration

Randomized SVD

- Orthogonal iteration for SVD can also be viewed as a randomized algorithm

Generalized Nyström Algorithm

- The generalized Nyström algorithm provides an efficient way of computing a sketched low-rank factorization

↙ sample size

S - sketch matrix , $n \times k$

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

- Gaussian random ← will densify

$O(n^2 k)$

- SRFT $S = \backslash\, D_n\, P$

diagonal random FFT permutation

$O(n^2 \log n)$

efficient if sketched data dense

- for sparse A, want to sample, Countsketch
- leverage score sampling

$S_1 S_2$

$$\underbrace{AS_2^T}_{n \times k}\underbrace{(S_1 A S_2^T)^+}_{k \times k}\underbrace{S_1 A}_{k \times n} = \tilde{A}$$

Multidimensional Optimization

- Minimize $f(x)$

$x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$

• constrained or unconstrained

equality

inequality

constrained & nonlinear

Lagrangian ↓ interior point

unconstrained nonlinear optimization

↓ Newton's method

quadratic optimization

$\nabla f(x^*) = 0$ or KKT

nonlinear solve

↓ Newton's method

linear system

- Quadratic optimization $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$

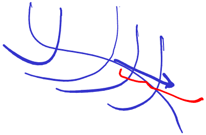
$\nabla f(x^*) = 0$

$Ax = b$

A is SPD

Basic Multidimensional Optimization Methods

- Steepest descent: minimize f in the direction of the negative gradient:



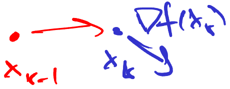
- Given quadratic optimization problem $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}$ where $\boldsymbol{A}$ is symmetric positive definite, the error $\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}^*$ satisfies

$$\underline{||\boldsymbol{e}_{k+1}||_{\boxed{\boldsymbol{A}}}} = e_{k+1}^T A e_{k+1} = ||e_k||_A \frac{\kappa(A)-1}{\kappa(A)+1}$$

  - When sufficiently close to a local minima, general nonlinear optimization problems are described by such an SPD quadratic problem.
  - Convergence rate depends on the conditioning of $\boldsymbol{A}$, since

Gradient Methods with Extrapolation

- We can improve the constant in the linear rate of convergence of steepest descent by leveraging *extrapolation methods*, which consider two previous iterates (maintain *momentum* in the direction $x_k - x_{k-1}$):



$$x_{k+1} - x_k = \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

- The *heavy ball method*, which uses constant $\alpha_k = \alpha$ and $\beta_k = \beta$, achieves better convergence than steepest descent:

$$\|e_{k+1}\|_A = \|e_k\|_A \frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}$$

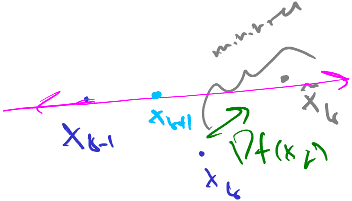
Nesterov's method

Conjugate Gradient Method

- The *conjugate gradient method* is capable of making the optimal (for a quadratic objective) choice of α_k and β_k at each iteration of an extrapolation method:

  - if A is $n \times n$, CG converges in n iterations
  - $x_{k+1} - x_k$ is A-orthogonal to prior directions

- *Parallel tangents* implementation of the method proceeds as follows



1st minimized $x_k + \alpha \nabla f(x_k) \rightarrow \bar{x}_k$

2nd minimized $x_{k-1} + \beta(\bar{x}_k - x_{k-1})$

Krylov Optimization

- Conjugate gradient (CG) finds the minimizer of $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$ (which satisfies optimality condition $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$) within the Krylov subspace of $\boldsymbol{A}$:

· each iteration of CG involves 1 matvec with A and vector ops

$$x_k \Leftarrow \min_{x \in \underbrace{\mathcal{K}_k(A,b)}_{\text{span}(Q_k)}} f(x) = \min_{y \in \mathbb{R}^k} \underbrace{\frac{1}{2} y^T \underbrace{Q_k^T A Q_k}_{T_k} y - y Q_k^T b}_{\varphi(y)}$$

$T_k \rightarrow$ is tridiagonal

$x = Q_k y$

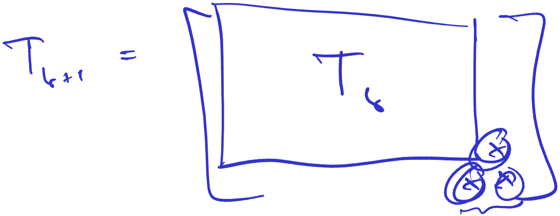
$e_1 \|b\|_2$

$$0 = \nabla \varphi(y) = \underbrace{Q_k^T A Q_k}_{T_k} y - \overbrace{Q_k^T b}^{e_1\|b\|_2}$$

$$T_k \underline{y} = e_1 \|b\|_2$$

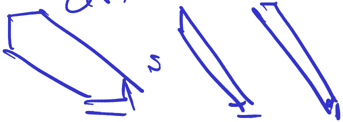
CG and Krylov Optimization

The solution at the kth step, $\boldsymbol{y}_k = ||\boldsymbol{b}||_2 \boldsymbol{T}_k^{-1} \boldsymbol{e}_1$ is obtained by CG from $\boldsymbol{y}_{k+1}$ with a single matrix-vector product with $\boldsymbol{A}$ and vector operations with $O(n)$ cost



$$y_{k+1} = ||b||_2 T_{k+1}^{-1} e_1$$

$O(k)$



$$\text{rank}\left(T_{k+1} - \begin{bmatrix} T_k & \\ & T_{k+1}(k+1,k+1) \end{bmatrix} \right) = 2$$

$$(M - uv^T)^{-1} = \left(M^{-1} + \frac{M^{-1}uv^TM^{-1}}{1 - v^TM^{-1}u} \right) \quad O(k)$$

Preconditioning

$\min_{x \in \ldots} \|Ax - b\|_{M^{-1}}$

- Convergence of iterative methods for $\boldsymbol{Ax} = \boldsymbol{b}$ depends on $\kappa(\boldsymbol{A})$, the goal of a preconditioner $\boldsymbol{M}$ is to obtain $\boldsymbol{x}$ by solving

$$\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b}$$

$M^{-1}A$

with $\kappa(\boldsymbol{M}^{-1}\boldsymbol{A}) < \kappa(\boldsymbol{A})$

never form $M^{-1}A$

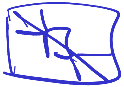
only apply A and solve with M

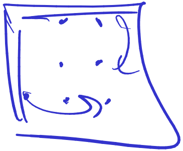
often, pick $M \approx A$ so that M^{-1} is easy to obtain/apply

$A = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \sigma_{\max} \ldots \sigma_{\min} \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}^T \to A - u_1 \sigma_{\max} v_1^T + u_1 v_1^T$

- Common preconditioners select parts of $\boldsymbol{A}$ or perform inexact factorization

• incomplete LU (LU without fill)





Conjugate Gradient Convergence Analysis

- In previous discussion, we assumed $\boldsymbol{K}_n$ is invertible, which may not be the case if $\boldsymbol{A}$ has $m < n$ distinct eigenvalues, however, in exact arithmetic CG converges in $m-1$ iterations[1]

[1] This derivation follows *Applied Numerical Linear Algebra* by James Demmel, Section 6.6.4

Conjugate Gradient Convergence Analysis (II)

- Using $\boldsymbol{z} = \boldsymbol{\rho}_{k-1}(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x}$, we can simplify $\phi(\boldsymbol{z}) = (\boldsymbol{x} - \boldsymbol{z})^T\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{z})$ as
- We can bound the objective based on the eigenvalues of $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ using the identity $\boldsymbol{p}(\boldsymbol{A}) = \boldsymbol{Q}\boldsymbol{p}(\boldsymbol{\Lambda})\boldsymbol{Q}^T$,

Conjugate Gradient Convergence Analysis (III)

- Using our bound on the square of the residual norm $\phi(\boldsymbol{z})$, we can see why CG converges after $m-1$ iterations if there are only $m<n$ distinct eigenvalues

- To see that the residual goes to 0, we find a suitable polynomial in $\mathcal{Q}_m$ (the set of polynomials q_m of degree m with $q_m(0)=1$)

Round-off Error in Conjugate Gradient

- CG provides strong convergence guarantees for SPD matrices in exact arithmetic

- Due to round-off CG may stagnate / have plateaus in convergence

Newton’s Method

- Newton’s method in n dimensions is given by finding minima of n-dimensional quadratic approximation using the gradient and Hessian of f:

Nonlinear Least Squares

- An important special case of multidimensional optimization is *nonlinear least squares*, the problem of fitting a nonlinear function $f_{\boldsymbol{x}}(t)$ so that $f_{\boldsymbol{x}}(t_i) \approx y_i$:

- We can cast nonlinear least squares as an optimization problem to minimize residual error and solve it by Newton's method:

Gauss-Newton Method

- The Hessian for nonlinear least squares problems has the form:

- The *Gauss-Newton* method is Newton iteration with an approximate Hessian:

Constrained Optimization Problems

- We now return to the general case of *constrained* optimization problems:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{h}(\boldsymbol{x}) \leqslant \boldsymbol{0}$$

- Generally, we will seek to reduce constrained optimization problems to a series of simpler optimization problems:

Lagrangian Duality

- The Lagrangian function with constraints $\boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$ and $\boldsymbol{h}(\boldsymbol{x}) \leqslant \mathbf{0}$ is

- The Lagrangian dual problem is an unconstrained optimization problem:

$$\max_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}), \quad q(\boldsymbol{\lambda}) = \begin{cases} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) & \text{if } \boldsymbol{\lambda} \geqslant \mathbf{0} \\ -\infty & \text{otherwise} \end{cases}$$

  The unconstrained optimality condition $\nabla q(\boldsymbol{\lambda}^*) = \mathbf{0}$, implies

Sequential Quadratic Programming

- *Sequential quadratic programming (SQP)* reduces a nonlinear equality constrained problem to a sequence of constrained quadratic programs via a Taylor expansion of the Lagrangian function $\mathcal{L}_f(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x})$:

- SQP ignores the constant term $\mathcal{L}_f(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$ and minimizes $\boldsymbol{s}$ while treating $\boldsymbol{\delta}$ as a Lagrange multiplier:

Interior Point Methods

- Barrier functions provide an effective way of working with inequality constraints $\boldsymbol{h}(x) \leqslant \mathbf{0}$:

- Interior point methods additionally incorporate Lagrangian optimization