CS 598 EVS: Tensor Computations

Matrix Computations Background

Edgar Solomonik

University of Illinois, Urbana-Champaign

Matrices and Tensors

- What is a matrix?

- What is a tensor?

Demo: Matrix norms

Matrix Norms

- **Properties of matrix norms**:

$$\begin{aligned} \|\boldsymbol{A}\| &\geqslant 0 \\ \|\boldsymbol{A}\| = 0 \quad &\Leftrightarrow \quad \boldsymbol{A} = \boldsymbol{0} \\ \|\alpha \boldsymbol{A}\| &= |\alpha| \cdot \|\boldsymbol{A}\| \\ \|\boldsymbol{A} + \boldsymbol{B}\| &\leqslant \|\boldsymbol{A}\| + \|\boldsymbol{B}\| \quad (\textit{triangle inequality}) \end{aligned}$$

- **Frobenius norm**:

- **Operator/induced/subordinate matrix norms**:

Existence of SVD

- **Consider any maximizer $\boldsymbol{x}_1 \in \mathbb{R}^n$ with $\|\boldsymbol{x}_1\|_2 = 1$ to $\|\boldsymbol{A}\boldsymbol{x}_1\|_2$**

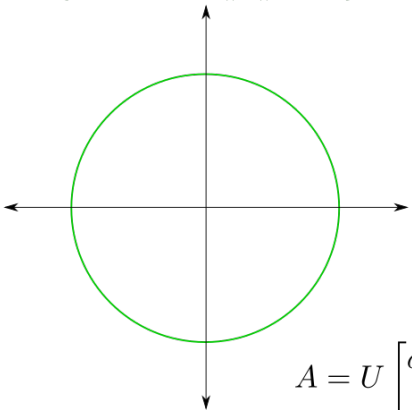
Singular Value Decomposition

- The singular value decomposition (SVD)

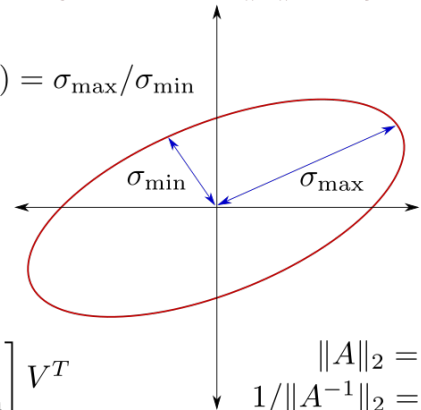
- Condition number in terms of singular values

Visualization of Matrix Conditioning

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



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



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

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

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:

Linear Systems

- Given a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with rank n, consider solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ given $\boldsymbol{b}$
- The SVD allows explicit inversion of $\boldsymbol{A}$

- However, Gaussian elimination is more computationally efficient

- Given a factorization of $\boldsymbol{A}$, solving a system with $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T$ has cost $O(n^2)$ via the Sherman-Morrison-Woodbury formula

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 $\dagger$ denotes pseudoinverse:

Normal Equations

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

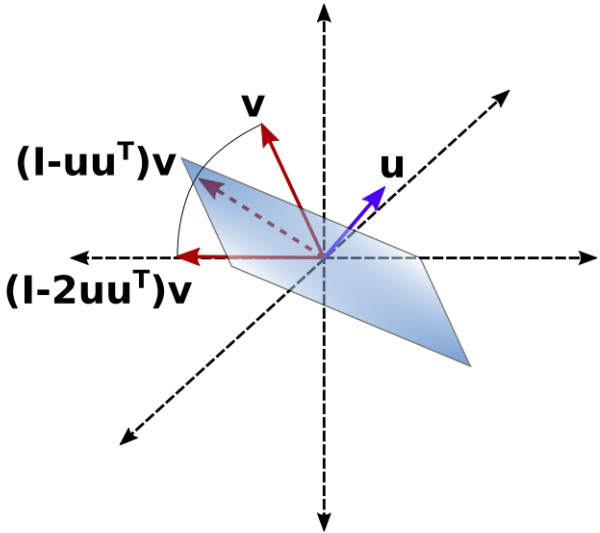
- We can solve the normal equations (and consequently the linear least squares problem) via reduced QR as follows

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

Householder orthogonalization



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

Invertible similarity transformations $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^{-1}$

matrix $(\boldsymbol{A})$	*reduced form* $(\boldsymbol{Y})$
arbitrary	
diagonalizable	

Unitary similarity transformations $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^{H}$

matrix $(\boldsymbol{A})$	*reduced form* $(\boldsymbol{Y})$
arbitrary	
normal	
Hermitian	

Orthogonal similarity transformations $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^{T}$

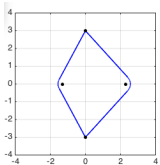
matrix $(\boldsymbol{A})$	*reduced form* $(\boldsymbol{Y})$
real	
real symmetric	
real SPD	

Field of Values

- For any square matrix $\boldsymbol{A}$ and vector $\boldsymbol{x}$ the *Rayleigh quotient* is
- Its magnitude is bounded by the singular values as
- If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$, so $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ or $\boldsymbol{x}^H\boldsymbol{A} = \lambda\boldsymbol{x}^H$, then
- The set $\mathcal{F}_{\boldsymbol{A}} = \{\rho_{\boldsymbol{A}}(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{C}^n, \boldsymbol{x} \neq 0\}$ is the *field of values* of $\boldsymbol{A}$

Field of Values and Eigenvalues

- Clearly any eigenvalue λ of $\boldsymbol{A}$ is in $\mathcal{F}_{\boldsymbol{A}}$
- For the matrix $\boldsymbol{A} = \begin{bmatrix} & 3 & & \\ -3 & & & \\ & & & 3 \\ & & 1 & 1 \end{bmatrix}$, $\mathcal{F}_{\boldsymbol{A}}$ is[1]



- The field of values of a normal matrix is easy to characterize

- In general, eigenvectors are obtained from critical points of the Rayleigh quotient on the unit circle

Singular Vectors as Critical Points

- Like eigenvectors, we can also derive singular vectors from an optimization (critical point) perspective

Matrix Functions

- Consider a polynomial p, for a diagonalizable matrix $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{-1}$,

$$p(\boldsymbol{A}) = \boldsymbol{X}p(\boldsymbol{D})\boldsymbol{X}^{-1}$$

- The above definition readily extends to other analytic functions f, but non-diagonalizable matrices require a more sophisticated definition

Crouzeix's conjecture

- So far, we have seen close connections between the matrix 2-norm and singular values, and between the Rayleigh quotient and the eigenvalues
- An important open problem in numerical analysis that relates the norm with the Rayleigh quotient is Crouzeix's conjecture

Computing Eigenvalue and Singular Value Decompositions

- Direct methods for eigenvalue problems start by reducing the matrix to upper-Hessenberg form

- Iterative methods are generally based on products with the matrix

Introduction to Krylov Subspace Methods

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

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

- Assuming $\boldsymbol{K}_n$ is invertible, the matrix $\boldsymbol{K}_n^{-1}\boldsymbol{A}\boldsymbol{K}_n$ is a *companion matrix* $\boldsymbol{C}$:

Krylov Subspaces

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

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

  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:

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 $\boldsymbol{Q}_k$:

Arnoldi Iteration

- Arnoldi iteration computes the ith column of $\boldsymbol{H}_n$, $\boldsymbol{h}_i$ and the ith column of $\boldsymbol{Q}_n$ directly using the recurrence $\boldsymbol{A}\boldsymbol{q}_i = \boldsymbol{Q}_n\boldsymbol{h}_i = \sum_{j=1}^{i+1} h_{ji}\boldsymbol{q}_j$

Multidimensional Optimization

- Minimize $f(\boldsymbol{x})$

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

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

  $$||\boldsymbol{e}_{k+1}||_{\boldsymbol{A}} =$$

  - 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 $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$):

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

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:

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

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 $\boldsymbol{A}$:

Conjugate Gradient Method: Optimized Form

After initialization $\boldsymbol{x}_0 = 0$, $\boldsymbol{r}_0 = \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$, the kth iteration of CG computes

$$\begin{aligned}
\boldsymbol{q}_k &= \boldsymbol{A}\boldsymbol{p}_k \\
\alpha_k &= \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{q}_k^T \boldsymbol{p}_k} \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k \\
\boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k \boldsymbol{q}_k
\end{aligned}$$

At this point if the residual norm ($\|\boldsymbol{r}_{k+1}\|$) is small, terminate, otherwise prepare for next iteration,

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \frac{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T \boldsymbol{r}_k} \boldsymbol{p}_k$$

See Jonathan Shewchuk 1994 notes on CG or James Demmel's book for the derivation of this form of the algorithm.

Conjugate Gradient Convergence Analysis

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

Round-off Error in Conjugate Gradient

- CG provides strong convergence guarantees for SPD matrices in exact arithmetic
  - Classically, CG was viewed as a direct method, since its guaranteed to convergence in n iterations
  - In practice, round-off error prevents CG from achieving this for realistic matrices, so CG was actually abandoned for a while due to being viewed as unstable
  - Later, it was realized that CG is highly competitive as an iterative method
- Due to round-off CG may stagnate / have plateaus in convergence
  - A formal analysis of round-off error[2] reveals that CG with round-off is equivalent to exact CG on a matrix of larger dimension, whose eigenvalues are clustered around those of $\boldsymbol{A}$
  - Using this view, CG convergence plateaus may be explained by the polynomial q_k developing more and more zeros near the same eigenvalue of $\boldsymbol{A}$

[2] A. Greenbaum and Z. Strakos, SIMAX 1992

Preconditioning

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

$$\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}$$

  with $\kappa(\boldsymbol{M}^{-1}\boldsymbol{A}) < \kappa(\boldsymbol{A})$
  - need not form $\boldsymbol{M}^{-1}\boldsymbol{A}$ but only compute matrix-vector products $\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x})$
  - want $\boldsymbol{M}^{-1}\boldsymbol{x}$ to be easy to compute (easier than $\boldsymbol{A}^{-1}\boldsymbol{x}$)
  - so generally one extracts some $\boldsymbol{M} \approx \boldsymbol{A}$ that is easy to solve linear systems with
  - however, $\boldsymbol{M} \approx \boldsymbol{A}$ may be insufficient/unnecessary, primary goal is to improve conditioning to accelerate iterative methods, i.e., want $\kappa(\boldsymbol{M}^{-1}\boldsymbol{A}) \ll \kappa(\boldsymbol{A})$
- Common preconditioners select parts of $\boldsymbol{A}$ or perform inexact factorization
  - (block-)Jacobi preconditioner takes $\boldsymbol{M}$ to be (block-)diagonal of $\boldsymbol{A}$
  - incomplete LU (ILU) preconditioners compute $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{U} \approx \boldsymbol{A}$ (+pivoting)
  - ILU variants constraint sparsity of $\boldsymbol{L}$ and $\boldsymbol{U}$ factors during factorization to be the same or not much more than that of $\boldsymbol{A}$
  - good problem-specific preconditioners are often available in practice and theory, applying also to problems beyond linear systems (eigenvalue problems, optimization, approximate graph algorithms)

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:

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

Simultaneous and Orthogonal Iteration

- Orthogonal iteration computing many eigenvectors at once in an iterative way

Orthogonal Iteration Convergence

- If $\boldsymbol{A}$ has distinct eigenvalues and $\boldsymbol{R}_i$ has positive decreasing diagonal, the jth column of $\boldsymbol{Q}_i$ converges to the jth Schur vector of $\boldsymbol{A}$ linearly with rate $\max(|\lambda_{j+1}/\lambda_j|, |\lambda_j/\lambda_{j-1}|)$.

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 low-rank factorization given an approximation of its span[3]

[3] Nakatsukasa, Yuji, Fast and stable randomized low-rank matrix approximation, 2020.

Analysis of Generalized Nyström Algorithm

- Consider $\boldsymbol{F}_1 = \boldsymbol{A}\boldsymbol{S}_1^T$ and $\boldsymbol{F}_2 = \boldsymbol{A}\boldsymbol{S}_2^T$, derive the minimizer $\boldsymbol{Z}$ to

$$\|\boldsymbol{A} - \boldsymbol{F}_1 \boldsymbol{Z} \boldsymbol{F}_2^T\|_F$$

- The generalized Nyström algorithm may be interpreted as applying a two-sided oblique projection of $\boldsymbol{A}$