CS 450: Numerical Anlaysis[1]

Linear Least Squares

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Linear Least Squares

- Find $\boldsymbol{x}^\star = \text{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$:
  Since $m \geq n$, the minimizer generally does not attain a zero residual $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$. We can rewrite the optimization problem constraint via

$$\boldsymbol{x}^\star = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{argmin}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{argmin}} \left[(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right]$$

- 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 $\boldsymbol{\Sigma}$:
  - *The minimizer satisfies $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}^\star \cong \boldsymbol{b}$ and consequently also satisfies*

$$\boldsymbol{\Sigma}\boldsymbol{y}^\star \cong \boldsymbol{d} \quad \textit{where } \boldsymbol{y}^\star = \boldsymbol{V}^T\boldsymbol{x}^\star \textit{ and } \boldsymbol{d} = \boldsymbol{U}^T\boldsymbol{b}.$$

  - *The minimizer of the reduced problem is $\boldsymbol{y}^\star = \boldsymbol{\Sigma}^\dagger\boldsymbol{d}$, so $y_i = d_i/\sigma_i$ for $i \in \{1,\ldots,n\}$ and $y_i = 0$ for $i \in \{n+1,\ldots,m\}$.*

Data Fitting via Linear Least Squares

- Given a set of m points with coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$, seek an $n-1$ degree polynomial p so that $p(x_i) \approx y_i$ by minimizing

$$\sum_{i=1}^{m}(y_i - p(x_i))^2 = \sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{n} z_j x_i^{j-1}\right)^2$$

where $\boldsymbol{z} \in \mathbb{R}^n$ *are the unknown polynomial coefficients*

- we can write this objective as a linear least squares problem

$$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2^2 \quad \textit{where} \quad \boldsymbol{A} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix}$$

Conditioning of Linear Least Squares

- Consider a perturbation $\boldsymbol{\delta b}$ to the right-hand-side $\boldsymbol{b}$

$$\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) \cong \boldsymbol{b} + \boldsymbol{\delta b}$$

- The amplification in relative perturbation magnitude (from $\boldsymbol{b}$ to $\boldsymbol{x}$) depends on how much of $\boldsymbol{b}$ is spanned by the columns of $\boldsymbol{A}$,

$$\begin{aligned}
(\boldsymbol{x} + \boldsymbol{\delta x}) &= \boldsymbol{A}^{\dagger}(\boldsymbol{b} + \boldsymbol{\delta b}) \\
\boldsymbol{\delta x} &= \boldsymbol{A}^{\dagger}\boldsymbol{\delta b} \\
\frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2} &= \frac{\|\boldsymbol{A}^{\dagger}\boldsymbol{\delta b}\|_2}{\|\boldsymbol{x}\|_2} \\
&\leq \frac{1}{\sigma_{min}(\boldsymbol{A})} \frac{\|\boldsymbol{\delta b}\|_2}{\|\boldsymbol{x}\|_2} \\
&\leq \frac{1}{\sigma_{min}(\boldsymbol{A})} \frac{\|\boldsymbol{\delta b}\|_2}{\|\boldsymbol{A x}\|_2 / \|\boldsymbol{A}\|_2} \\
&\leq \kappa(\boldsymbol{A}) \frac{\|\boldsymbol{b}\|_2}{\|\boldsymbol{A x}\|_2} \frac{\|\boldsymbol{\delta b}\|_2}{\|\boldsymbol{b}\|_2}
\end{aligned}$$

Normal Equations

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

- *Normal equations* are given by solving $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$:

  If $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$ then

$$\begin{aligned}
(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T\boldsymbol{b} \\
\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} &= \boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{b} \\
\boldsymbol{V}^T\boldsymbol{x} &= (\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{b} = \boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T\boldsymbol{b} \\
\boldsymbol{x} &= \boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T\boldsymbol{b} = \boldsymbol{x}^\star
\end{aligned}$$

- However, solving the normal equations is a more ill-conditioned problem then the original least squares algorithm

  Generally we have $\kappa(\boldsymbol{A}^T\boldsymbol{A}) = \kappa(\boldsymbol{A})^2$ (the singular values of $\boldsymbol{A}^T\boldsymbol{A}$ are the squares of those in $\boldsymbol{A}$). Consequently, solving the least squares problem via the normal equations may be unstable because it involves solving a problem that has worse conditioning than the initial least squares problem.

Solving the Normal Equations

- If $\boldsymbol{A}$ is full-rank, then $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric positive definite (SPD):
  - *Symmetry is easy to check* $(\boldsymbol{A}^T\boldsymbol{A})^T = \boldsymbol{A}^T\boldsymbol{A}$*.*
  - $\boldsymbol{A}$ *being full-rank implies* $\sigma_{min} > 0$ *and further if* $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ *we have*

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{V}^T\boldsymbol{\Sigma}^2\boldsymbol{V}$$

    which implies that rows of $\boldsymbol{V}$ *are the eigenvectors of* $\boldsymbol{A}^T\boldsymbol{A}$ *with eigenvalues* $\boldsymbol{\Sigma}^2$ *since* $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}^T = \boldsymbol{V}^T\boldsymbol{\Sigma}^2$*.*

- Since $\boldsymbol{A}^T\boldsymbol{A}$ is SPD we can use Cholesky factorization, to factorize it and solve linear systems:

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$$

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 $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$
  - *Given $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$, we can take $\boldsymbol{R} = \boldsymbol{L}^T$ and obtain $\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{L}^{-T}$, since $\underbrace{\boldsymbol{L}^{-1}\boldsymbol{A}^T}_{\boldsymbol{Q}^T}\underbrace{\boldsymbol{A}\boldsymbol{L}^{-T}}_{\boldsymbol{Q}} = \boldsymbol{I}$ implies that $\boldsymbol{Q}$ has orthonormal columns.*
- 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

  A full QR factorization gives $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{R} \in \mathbb{R}^{m\times n}$, but since $\boldsymbol{R}$ is upper triangular, the latter $m-n$ columns of $\boldsymbol{Q}$ are only constrained so as to keep $\boldsymbol{Q}$ orthogonal. The reduced QR factorization is given by taking the first n columns $\boldsymbol{Q}$ and $\hat{\boldsymbol{Q}}$ the upper-triangular block of $\boldsymbol{R}$, $\hat{\boldsymbol{R}}$ giving $\boldsymbol{A} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}$.
- We can solve the normal equations (and consequently the linear least squares problem) via reduced QR as follows

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b} \quad \Rightarrow \quad \hat{\boldsymbol{R}}^T\underbrace{\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}}_{\boldsymbol{I}}\hat{\boldsymbol{R}}\boldsymbol{x} = \hat{\boldsymbol{R}}^T\hat{\boldsymbol{Q}}^T\boldsymbol{b} \quad \Rightarrow \quad \hat{\boldsymbol{R}}\boldsymbol{x} = \hat{\boldsymbol{Q}}^T\boldsymbol{b}$$

Gram-Schmidt Orthogonalization

***Demo:** Gram-Schmidt–The Movie*
***Demo:** Gram-Schmidt and Modified Gram-Schmidt*

- **Classical Gram-Schmidt process for QR:**

  The Gram-Schmidt process orthogonalizes a rectangular matrix, i.e. it finds a set of orthonormal vectors with the same span as the columns of the given matrix. If $\boldsymbol{a}_i$ is the ith column of the input matrix, the ith orthonormal vector (ith column of $\boldsymbol{Q}$) is

$$\boldsymbol{q}_i = \boldsymbol{b}_i / \underbrace{||\boldsymbol{b}_i||_2}_{r_{ii}}, \quad \textit{where} \quad \boldsymbol{b}_i = \boldsymbol{a}_i - \sum_{j=1}^{i-1} \underbrace{\langle \boldsymbol{q}_j, \boldsymbol{a}_i \rangle}_{r_{ji}} \boldsymbol{q}_j .$$

- **Modified Gram-Schmidt process for QR:**

  Better numerical stability is achieved by orthogonalizing each vector with respect to each previous vector in sequence (modifying the vector prior to orthogonalizing to the next vector), so $\boldsymbol{b}_i = \mathit{MGS}(\boldsymbol{a}_i, i-1)$, where $\mathit{MGS}(\boldsymbol{d}, 0) = \boldsymbol{d}$ and

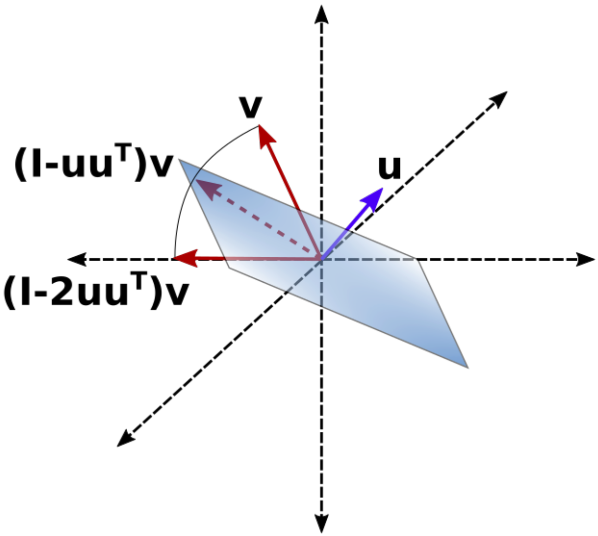
$$\mathit{MGS}(\boldsymbol{d}, j) = \mathit{MGS}(\boldsymbol{d} - \langle \boldsymbol{q}_j, \boldsymbol{d} \rangle \boldsymbol{q}_j, j-1)$$

Householder QR Factorization

- **A Householder transformation $\boldsymbol{Q} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T$ is an orthogonal matrix defined to annihilate entries of a given vector $\boldsymbol{z}$, so $\boldsymbol{Q}\boldsymbol{z} = \pm||\boldsymbol{z}||_2\boldsymbol{e}_1$:**
  - *Householder QR achieves unconditional stability, by applying only orthogonal transformations to reduce the matrix to upper-triangular form.*
  - *Householder transformations (reflectors) are orthogonal matrices, that reduce a vector to a multiple of the first elementary vector, $\alpha\boldsymbol{e}_1 = \boldsymbol{Q}\boldsymbol{z}$.*
  - *Because multiplying a vector by an orthogonal matrix preserves its norm, we must have that $|\alpha| = ||\boldsymbol{z}||_2$.*
  - *As we will see, this transformation can be achieved by a rank-1 perturbation of identify of the form $\boldsymbol{Q} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T$ where $\boldsymbol{u}$ is a normalized vector.*
  - *Householder matrices are both symmetric and orthogonal implying that $\boldsymbol{Q} = \boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$.*
- Imposing this form on $\boldsymbol{Q}$ leaves exactly two choices for $\boldsymbol{u}$ given $\boldsymbol{z}$,

$$\boldsymbol{u} = \frac{\boldsymbol{z} \pm ||\boldsymbol{z}||_2\boldsymbol{e}_1}{||\boldsymbol{z} \pm ||\boldsymbol{z}||_2\boldsymbol{e}_1||_2}$$

Visualization of Householder Reflector



Applying Householder Transformations

- The product $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{w}$ can be computed using $O(n)$ operations if $\boldsymbol{Q}$ is a Householder transformation

$$\boldsymbol{x} = (\boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{w} = \boldsymbol{w} - 2\langle \boldsymbol{u}, \boldsymbol{w}\rangle \boldsymbol{u}$$

- Householder transformations are also called *reflectors* because their application reflects a vector along a hyperplane (changes sign of component of $\boldsymbol{w}$ that is parallel to $\boldsymbol{u}$)
  - *$\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^T$ would be an elementary projector, since $\langle \boldsymbol{u}, \boldsymbol{w}\rangle \boldsymbol{u}$ gives component of $\boldsymbol{w}$ pointing in the direction of $\boldsymbol{u}$ and*

    $$\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{w} = \boldsymbol{w} - \langle \boldsymbol{u}, \boldsymbol{w}\rangle \boldsymbol{u}$$

    subtracts it out.
  - *On the other hand, Householder reflectors give*

    $$\boldsymbol{y} = (\boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{w} = \boldsymbol{w} - 2\langle \boldsymbol{u}, \boldsymbol{w}\rangle \boldsymbol{u} = \boldsymbol{x} - \langle \boldsymbol{u}, \boldsymbol{w}\rangle \boldsymbol{u}$$

    which reverses the sign of that component, so that $||\boldsymbol{y}||_2 = ||\boldsymbol{w}||_2$.

Givens Rotations

- Householder reflectors reflect vectors, Givens rotations rotate them
  - *Householder matrices reflect vectors across a hyperplane, by negating the sign of the vector component that is perpendicular to the hyperplane (parallel to $\boldsymbol{u}$)*
  - *Any vector can be reflected to a multiple of an elementary vector by a single Householder rotation (in fact, there are two rotations, resulting in a different sign of the resulting vector)*
  - *Givens rotations instead rotate vectors by an axis of rotation that is perpendicular to a hyperplane spanned by two elementary vectors*
  - *Consequently, each Givens rotation can be used to zero-out (annihilate) one entry of a vector, by rotating it so that the component of the vector pointing in the direction of the axis corresponding to that entry, points into a different axis*
- Givens rotations are defined by orthogonal matrices of the form $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$
  - *Given a vector* $\begin{bmatrix} a \\ b \end{bmatrix}$ *we define* c *and* s *so that* $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{a^2+b^2} \\ 0 \end{bmatrix}$
  - *Solving for* c *and* s*, we get* $c = \frac{a}{\sqrt{a^2+b^2}}, \quad s = \frac{b}{\sqrt{a^2+b^2}}$

QR via Givens Rotations

- We can apply a Givens rotation to a pair of matrix rows, to eliminate the first nonzero entry of the second row

$$\begin{bmatrix} \boldsymbol{I} & & & & \\ & c & & s & \\ & & \boldsymbol{I} & & \\ & -s & & c & \\ & & & & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \vdots \\ a \\ \vdots \\ b \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \sqrt{a^2+b^2} \\ \vdots \\ 0 \\ \vdots \end{bmatrix}$$

- Thus, $n(n-1)/2$ Givens rotations are needed for QR of a square matrix
  - *Each rotation modifies two rows, which has cost* $O(n)$
  - *Overall, Givens rotations cost* $2n^3$*, while Householder QR has cost* $(4/3)n^3$
  - *Givens rotations provide a convenient way of thinking about QR for sparse matrices, since nonzeros can be successively annihilated, although they introduce the same amount of fill (new nonzeros) as Householder reflectors*

Rank-Deficient Least Squares

- Suppose we want to solve a linear system or least squares problem with a (nearly) rank deficient matrix $\boldsymbol{A}$
  - *A rank-deficient (singular) matrix satisfies $\boldsymbol{Ax} = \boldsymbol{0}$ for some $\boldsymbol{x} \neq \boldsymbol{0}$*
  - *Rank-deficient matrices must have at least one zero singular value*
  - *Matrices are said to be deficient in numerical rank if they have extremely small singular values*
  - *The solution to both linear systems (if it exists) and least squares is not unique, since we can add to it any multiple of $\boldsymbol{x}$*
- Rank-deficient least squares problems seek a minimizer $\boldsymbol{x}$ of $||\boldsymbol{Ax} - \boldsymbol{b}||_2$ of minimal norm $||\boldsymbol{x}||_2$
  - *If $\boldsymbol{A}$ is a diagonal matrix (with some zero diagonal entries), the best we can do is $x_i = b_i/a_{ii}$ for all i such that $a_{ii} \neq 0$ and $x_i = 0$ otherwise*
  - *We can solve general rank-deficient systems and least squares problems via $\boldsymbol{x} = \boldsymbol{A}^\dagger \boldsymbol{b}$ where the pseudoinverse is*

$$\boldsymbol{A}^\dagger = \boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T \quad \sigma_i^\dagger = \begin{cases} 1/\sigma_i & : \sigma_i > 0 \\ 0 & : \sigma_i = 0 \end{cases}$$

Truncated SVD

- After floating-point rounding, rank-deficient matrices typically regain full-rank but have nonzero singular values on the order of $\epsilon_{\text{mach}}\sigma_{\text{max}}$
  - *Very small singular values can cause large fluctuations in the solution*
  - *To ignore them, we can use a pseudoinverse based on the truncated SVD which retains singular values above an appropriate threshold*
  - *Alternatively, we can use Tykhonov regularization, solving least squares problems of the form* $\min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 + \alpha||\boldsymbol{x}||_2^2$*, which are equivalent to the augmented least squares problem*

$$\begin{bmatrix} \boldsymbol{A} \\ \sqrt{\alpha}\boldsymbol{I} \end{bmatrix} \boldsymbol{x} \cong \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}$$

- By the *Eckart-Young-Mirsky theorem*, truncated SVD also provides the best low-rank approximation of a matrix (in 2-norm and Frobenius norm)
  - *The SVD provides a way to think of a matrix as a sum of outer-products* $\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ *that are disjoint by orthogonality and the norm of which is* σ_i
  - *Keeping the* r *outer products with largest norm provides the best rank-*r *approximation*

QR with Column Pivoting

- QR with column pivoting provides a way to approximately solve rank-deficient least squares problems and compute the truncated SVD
  - *We seek a factorization of the form* $\boldsymbol{QR} = \boldsymbol{AP}$ *where* $\boldsymbol{P}$ *is a permutation matrix that permutes the columns of* $\boldsymbol{A}$
  - *For* $n \times n$ *matrix* $\boldsymbol{A}$ *of rank* r*, the bottom* $r \times r$ *block of* $\boldsymbol{R}$ *will be* $\mathbf{0}$
  - *To solve least squares, we can solve the rank-deficient triangular system* $\boldsymbol{Ry} = \boldsymbol{Q}^T\boldsymbol{b}$ *then compute* $\boldsymbol{x} = \boldsymbol{Py}$
- A pivoted QR factorization can be used to compute a rank-r approximation
  - *To compute QR with column pivoting,*
    1. *pivot the column of largest norm to be the leading column,*
    2. *form and apply a Householder reflector* $\boldsymbol{H}$ *so that* $\boldsymbol{HA} = \begin{bmatrix} \alpha & \boldsymbol{b} \\ \mathbf{0} & \boldsymbol{B} \end{bmatrix}$,
    3. *proceed recursively (go back to step 1) to pivot the next column and factorize* $\boldsymbol{B}$
  - *Computing the SVD of the first* r *columns of* $\boldsymbol{AP}^T$ *gives approximations that are typically almost as good as the truncated SVD, but other "rank-revealing" QR algorithms exist with more robust guarantees*
  - *Halting after* r *steps leads to a cost of* $O(n^2r)$