1 Linear algebra review

1.1 Vector products

$x^T y = \sum x_i * y_i$. $x^T y = \|x\|_2 \|y\|_2 \cos\theta$. $x^T y = 0 \Leftrightarrow x \perp y$

1.2 Norms

Measures of the length of vectors and matrices. All norms satisfy

- Only zero vector has zero norm: $\|x\|_x = 0 \Leftrightarrow x = 0$
- $\|\alpha x\|_x = |\alpha| \|x\|_x$
- $\|x + y\|_x \leq \|x\|_x + \|y\|_x$ (Triangle inequality) (also $\|x - y\|_x \geq \|x\|_x - \|y\|_x$)

1.2.1 Vector norms

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- $\|x\|_2 = \sqrt{\sum_{i=1}^{n} (x_i)^2}$
- $\|x\|_\infty = \max_{i \in i,\dots,n} |x_i|$
- $\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$

Cauchy-Schwarts Inequality: $|x^T y| \leq \|x\|_2 \|y\|_2$ (note equality when $x^T y = 0$)

Holder's Inequality: $|x^T y| \leq \|x\|_p \|y\|_q$, for p, q , s.t. $\frac{1}{p} + \frac{1}{q} = 1$

1.2.2 Matrix norms

Types of **matrix norms**, $A \in \mathbb{R}^{n \times m}$

- $\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_{\|x\|_\infty = 1} \|Ax\|_\infty = \max_i \left\|a_i^T\right\|_1$
- $\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|Ax\|_p$
- $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{tr(AA^T)} = \sqrt{tr(A^T A)} = \sqrt{\sum_{k=1}^{min(m,n)} \sigma_k^2}$

Submultiplicative inverse: $\|AB\|_p \leq \|A\|_p \|B\|_p$. Note: this is not always true for Frobenius norms.

Induced 2-norm: $\|Ay\|_p \leq \|A\|_p \|y\|_p$

Orthogonally invariant: Orthogonal matrices do not change the norms of vectors or matrices:

- $\|Qx\|_x = \|x\|_p$
- $\|QA\|_x = \|A\|_x, x \in \{p, F\}$

Other norm properties

- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty$
- $\|A\|_2 \leq \sqrt{m} \|A\|_\infty$
- $\|A\|_\infty \leq \sqrt{n} \|A\|_2$

1.3 Matrix properties

1.3.1 Determinant

Determinant represents how volume of hypercube is tranformed by a matrix.

- For square matrix, $det(\alpha A) = \alpha^n det(A)$
- For square matrices, $det(AB) = det(A)det(B)$
- $det(A) = det(A^T)$
- $det(A^{-1}) = \frac{1}{det(A)}$
- For square matrix, A singular $\Leftrightarrow det(A) = 0 \Leftrightarrow$ columns of A are not linearly independent

1.3.2 Trace

$tr(A) = \sum_{i=1}^{n} a_{ii}$

- $tr(A) = tr(A^T)$
- $tr(A + \alpha B) = tr(A) + \alpha tr(B)$
- Trace is invariant under cyclic permutations, that is $tr(ABCD) = tr(BCDA) = tr(CDAB) = tr(DABC)$
- For two vectors, $u, v \in \mathbb{R}, tr(uv^T) = v^T u$

1.3.3 Inverses and transposes

The inverse of the transpose is the transpose of the inverse: $A^T(A^{-1})^T = (A^{-1}A)^T = I^T = I = I^T = (AA^{-1})^T = (A^{-1})^T A^T$

1.3.4 Sherman-Morrison-Woodbury formula

for $A \in \mathbb{R}^{n\times n}, U, V \in \mathbb{R}^{n\times k}$ $(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}$
Proof: begin with the inverse of the LHS multiplied by the RHS: $(A + UV^T)(A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1})$. Next perform matrix multiplication. The end result will be I

1.4 Orthogonal matrices

An orthogonal matrix, Q is a matrix whose columns are orthonormal. That is $q_i^T q_j = 1$ for $i = j$, $q_i^T q_j = 0$ for $i \neq j$. Equivalently, $Q^T Q = I$. For square matrices, $Q^T Q = QQ^T = I$

1.5 Projections, reflections, and rotations

1.5.1 Projections

A projection, v, of vector x onto vector y can be written in the form $v = \frac{y^T x}{y^T y} y$. Which can be interpreted as the portion of x in the direction of y ($y^T x$), times the direction of y, divided by the length of y twice ($y^T y = \|y\|_2^2$), since y appears in the dot product and in the vector.
Projection matrices are square matrices, P, s.t., $P^2 = P$.

1.5.2 Reflection

P is a reflection matrix $\Leftrightarrow P^2 = I$. P can be written in the form $P = I - \beta vv^T$, with $\beta = \frac{2}{v^T v}$, and v the vector orthogonal to the line/plane of reflection. It can be shown that $Px = x \Leftrightarrow v^T x = 0$. These x are called the "fixed points" of P.

1.6 Symmetric Positive Definite (SPD) Matrices

For A, SPD, $A = A^T$, $x^T Ax > 0\ \forall x \neq 0$, $a_{ii} > 0$, $\lambda(A) \geq 0$. And for B nonsingular, $B^T AB$ is also SPD

When proving properties of SPDs, use the following tricks: i) Multiply by e_i since $e_i \neq 0$. Use matrix transpose property, $x^T A^T = (Ax)^T$ to rearrange formulas

1.6.1 $B^T AB$ is also SPD

If A SPD $\Rightarrow B^T AB$ SPD for B nonsingular: $x^T B^T ABx = (Bx)^T A(Bx) > 0$, (since B nonsingular $\Rightarrow Bx \neq 0$)

1.7 Eigenvalues

Observe by definition $Ax = \lambda x \Rightarrow (A - \lambda I)x = 0$. To find lambda, we solve for the system of equations to satisfy $(A - \lambda I)x = 0$. Also, $\lambda(A) = \lambda(A^T)$

The **algebraic multiplicity** of an eigenvalue, λ_i, is the number of times that the value λ_i appears as an eigenvalue of A e.g., for characteristic equation $p(x) = (x-2)^3(x-1)^2$, $\lambda = 2$ has algebraic multiplicity of 3
The **geometric multiplicity** of an eigenvalue, λ_i, is the dimension of the space spanned by the eigenvectors of λ_i

1.7.1 Determinants and trace

$det(A) = \prod_{i=1}^{n} \lambda_i$; $tr(A) = \sum_{i=1}^{n} \lambda_i$

1.7.2 Triangular matrices

For T triangular, the eigenvalues appear on the diagonal: $t_{ii} = \lambda_i, \forall i \in \{1, \ldots, n\}$. **Corollary:** T nonsingular $\Leftrightarrow$ all $t_{ii} \neq 0$

1.7.3 Gershgorin disc theorem

Gershgorin disc, $\mathbb{D}_i = \mathbb{D}_i = \{z \in \mathbb{C} \mid |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}$. All eigenvalues of A, $\lambda(A) \in \mathbb{C}$ are located in one of its Gershgorin discs
Proof:

$$\begin{aligned}
(A - \lambda I)x &= 0 \\
\sum_{j \neq i} a_{ij} x_j + (a_{ii} - \lambda) x_i &= 0, \ \forall i \in \{1, \ldots, n\} \\
\text{Choose } i \ s.t. |x_i| &= \max_i |x_i| \\
|(a_{ii} - \lambda)| &= |\sum_{j \neq i} \frac{a_{ij} x_j}{x_i}| \leq \sum_{j \neq i} |\frac{a_{ij} x_j}{x_i}| \text{ , by triangle inequality} \\
|(\lambda - a_{ii})| &\leq \sum_{j \neq i} |a_{ij}|, \text{ since } |\frac{x_j}{x_i}| \leq 1
\end{aligned}$$

2 Matrix Decompositions

2.1 Schur Decomposition

- For any $A \in \mathbb{C}^{n \times n}$, $A = QTQ^H$, where Q unitary $(Q^H Q = I), Q \in \mathbb{C}^{n \times n}$, T upper triangular
- When $A \in \mathbb{R}^{n \times n}$, the Real Schur Decomposition becomes $A = QTQ^T$, where Q orthogonal $(Q^T Q = I), Q \in \mathbb{R}^{n \times n}$, T upper triangular
- Note: If T is relaxed from strict upper triangular to block upper triangular (blocks of 2×2 or 1×1 on the diagonal), then Q can be selected to be in $\mathbb{R}^{n \times n}$.

2.2 Eigenvalue Decomposition

- For A diagonalizable ($A \in \mathbb{R}^{n \times n}$ with n linearly independent eigenvectors), it can be decomposed as $A = X \Lambda X^{-1}$, where Λ a diagonal matrix of the eigenvalues of A
- For A real symmetric, A can be decomposed as $A = Q \Lambda Q^T, Q$ orthogonal
- For A unitarily diagonalizable ($\Leftrightarrow$ normal: $A^H A = AA^H$), A can be decomposed as $A = Q \Lambda Q^H, Q$ unitary. When A complex Hermitian $(A = A^H)$, $\Lambda \in \mathbb{R}$

2.3 Singular Value Decomposition

Definition: For any $A \in \mathbb{C}^{m \times n}$ there exist two unitary matrices, $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$, and a diagonal matrix $\Sigma \in \mathbb{R}^{m \times n}$ such that $A = U \Sigma V^H$. When $A \in \mathbb{R}^{m \times n}, A = U \Sigma V^T$ with $U, V, \Sigma \in \mathbb{R}$.

The singular values, σ_i of Σ are always ≥ 0. And by convention, they're ordered in decreasing order, so $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Derivation: Observe A^TA symmetric: $(A^TA)^T = A^TA$

A^TA symmetric $\Rightarrow \exists\, Q$ orthogonal and Λ diagonal matrix of λ_i s.t.,

$$A^TA = Q\Lambda Q^T \Rightarrow Q^TA^TAQ = Q^TQ\Lambda Q^TQ$$

$(AQ)^T(AQ) = \Lambda$, note AQ is orthogonal, but not scaled to 1. Instead, each row is scaled to the eigenvalue in that row: $\lambda_i = \|Aq_i\|_2^2$

When A is full rank,

$$A = AQQ^T = (AQ)Q^T = AQD^{-1}DQ^T, \text{ where } D = \begin{bmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \sqrt{\lambda_n} \end{bmatrix} \text{ and } D^{-1} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \frac{1}{\sqrt{\lambda_n}} \end{bmatrix}$$

$$A = U\Sigma V^T, \text{ where } U = AQD^{-1}, \Sigma = D, V^T = Q^T$$

When A is not full rank, this does not hold since $\lambda_i = 0$ for some i so we cannot construct U with D^{-1}. The trick in this case is to construct a tall/thin AQ, and a D with $\sqrt{\lambda_i}$ where nonzero in the upper block, and an Identity matrix in the lower block. And a few additional properties and remarks of $A \in \mathbb{R}^{n\times m}$ SVD

- $\|A\|_2 = \sigma_1$; $\left\|A^{-1}\right\|_2 = \frac{1}{\sigma_n}$ when A nonsingular
- $\|A\|_F = \sqrt{\sum_i^{min\{n,m\}} \sigma_i^2}$
- When A symmetric, $\sigma_i = |\lambda_i|$. When A orthogonal, $\sigma_1 = \dots = \sigma_n = 1$
- The eigenvalues of A^TA and AA^T are the squares of the singular values of A, $\sigma_1^2, \dots, \sigma_n^2$
- By construction, V contains the eigenvectors of A^TA and U contains the eigenvectors of AA^T, so $A^TAv_i = \sigma_i^2 v_i$ and $AA^Tu_i = \sigma_i^2 u_i$
- **Condition number**, $\kappa(A) = \|A\|_2\left\|A^{-1}\right\|_2 = \frac{\sigma_1}{\sigma_n}$

3 Error analysis

3.1 Floating point arithmetic

General floating point number equation: $\pm(\sum_{i=1}^{t-1} d_i\beta^{-i})\beta^e$
Where β is the base (in floating point computation, $\beta = 2$). $d_0 \geq 1$, and $d_i \leq \beta - 1$. e is called the **exponent**, this is the location of the decimal place. $t-1$ in the summand is called the **precision** and indicates the number of digits (in base β) that can be stored with the number. The part of the equation in the parenthesis is referred to as the **significand** or **mantissa**

3.2 Unit roundoff

The **unit roundoff** for a floating-point number is

$$u = \frac{1}{2} \times \beta^{-(t-1)} \text{ (distance between the smallest digits stored in a floating-point number)}$$

For double precision floating point numbers (64 bits) $u \approx 10^{-16}$
The **floating point truncation operator**, $fl(a)$, takes as input a and returns the nearest floating point, $fl(a)$. Observe

$$fl(a+b) = a + b + \epsilon(a+b),\ |\epsilon| \leq u, \text{ the unit roundoff}$$

3.3 Forward/Backward error analysis

Forward error analysis The forward error is $\left\|\tilde{f}(x) - f(x)\right\|_p$. i.e., What is the error in the solution computed with our algorithm?

Backward error analysis is $\tilde{E}$ such that $(A + \tilde{E})\tilde{x} = b$. i.e., what is the problem that our algorithm actually solved? is regarded as *backward stable* if $\|E\|_p \in O(u)$

The relative sensitivity of a problem is often called the **conditioning** of the problem. **Sensitivity:** $\frac{\left\|\tilde{f}(x)-f(x)\right\|_p}{\|\tilde{x}-x\|_p}$. **Relative sensitivity:** $\frac{\left\|\tilde{f}(x)-f(x)\right\|_p\|x\|_p}{\|\tilde{x}-x\|_p\|f(x)\|_p}$

4 LU Factorization

Once we have $A = LU$, to solve $Ax = b$, we can start by solving $Lz = b$, and then $Ux = z$. x, here, is the solution!

4.1 Basic algorithm

Iteratively subtracting the outer products of vectors that sequentially "zero-out" the rows and columns of A. $LU = l_1 u_1^t + \cdots + l_n u_n^t$. $LU - l_1 u_1^t$ yields a matrix with zeros in the first row and column. We use this principle for the basic algorithm

- Construct u_1^T equal to the first row of A, a_1^T
- Construct l_1 equal to each of the elements in the first column of A, a_1, divided by a_{11}, the "pivot"
- Calculate $A' \leftarrow A - l_1 u_1^T$. In practice (and somewhat confusingly), A' is now referred to as A
- Repeat the algorithm with the updated A, and the next row/column. Observe each l_i, u_i^T constructed are the rows/columns of the lower and upper triangular matrices of L, U respectively.

4.1.1 Gauss transforms

Another way to think about the basic LU factorization algorithm is with Gauss transforms. **Guass transformation matrices** are linear transformations that zero out all entries below a certain entry. The columns of a Gauss transformation look like the values of l_i, where nonzero entries are divided by a pivot entry.

4.2 Pivoting

4.2.1 When pivoting is needed

a_{kk}, being nonzero if none of the $k \times k$ blocks of A, $A[1:k, 1:k]$, have a determinant of 0. **Proof by induction**:
Case k=1:

$$\begin{aligned}
A_1 &= L_1 U_1 \\
det(A_1) &= det(L_1 U_1) \\
det(A_1) &= det(L_1)det(U_1), \text{ by property of determinants} \\
det(A_1) &= det(U_1), \text{ since determinant of a triangular matrix is a product of the diagonals and the diagonal of } L_1 \text{ are 1's} \\
det(A_1) &= a_{11} = u_{11} \rightarrow \text{so when determinant is not zero, we have a nonzero pivot}
\end{aligned}$$

Case k=n: assumed to be true
Case k=n+1: $det(A_{k+1}) = u_{11} * u_{22} * \cdots * u_{kk}$ but we know $u_{ii} \neq 0$ for $i \leq k$ from induction step, so when determinant is not zero, we have pivot, $a_{k+1,k+1}$ nonzero.
What's more, if the entries of L are large (which occurs when entries in A are really small and land on the pivot locations), then because of roundoff errors in a computer, this algorithm can generate errors.

4.2.2 Pivoting algorithms

Pivoting algorithms pivot the iterative version of A in each iteration to avoid the numerical issues identified above. **Partial/Row pivoting** performs row swaps for max remaining entry and solves $PA = LU$. **Full pivoting** performs row and column swaps and solves $PAQ^T = LU$. Full pivoting is **rank-revealing**. **Rook pivoting** performs row and column swaps for max of row/col.

4.3 Cholesky factorization

The Cholesky factorization is an LU factorization for Symmetric Positive Definite (SPD) matrices, where SPD matrix, $A = GG^T$, with G lower triangular.
Intuition: An SPD matrix, A, can be written of the form

$$A = \begin{bmatrix} a & C^T \\ C & B \end{bmatrix} \text{where a is 1x1, C is n-1x1, and b is n-1xn-1}$$

After the first step of the LU factorization, we have the following matrix product, $A = L_1U_1$

$$\begin{bmatrix} a & C^T \\ C & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C/a & I \end{bmatrix} \begin{bmatrix} a & C^T \\ 0 & B-(1/a)CC^T \end{bmatrix}$$

Notice since A is symmetric, B is also symmetric, so $B-(1/a)CC^T$ must by symmetric by construction. We are also guaranteed to have the pivot, a in entry $(1,1)$ of A, to be strictly greater than zero since A is SPD: $a = e_1^T A e_1 > 0$. Next, we can further decompose the second matrix to

$$A = \begin{bmatrix} 1 & 0 \\ C/a & I \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & B-(1/a)CC^T \end{bmatrix} \begin{bmatrix} 1 & C^T/a \\ 0 & I \end{bmatrix}$$

Using the fact that A SPD $\Rightarrow B^TAB$ SPD for B nonsingular, observe that matrix $\begin{bmatrix} 1 & 0 \\ C/a & I \end{bmatrix}$ is nonsingular so therefore the matrix $\begin{bmatrix} a & 0 \\ 0 & B-(1/a)CC^T \end{bmatrix}$ must be SPD. Which also means the submatrix $B-(1/a)CC^T$ is SPD
We can use induction to prove that the Cholesky factorization exists.

Continuing with this factorization, we get an equation of the form $A = LDL^T$ for D, diagonal, and L, lower triangular. It's common to rewrite $A = LDL^T$ in the form $A = GG^T$, where $G = LD^{\frac{1}{2}}$

4.3.1 Cholesky factorization is unique

By contradiction, suppose $A = GG^T = MM^T$ for $G \neq M$ We know G, M nonsingular (consider the determinants of the equation above) so

$$\begin{aligned}
GG^T &= MM^T \\
I &= G^{-1}MM^TG^{-T} \\
I &= (G^{-1}M)(G^{-1}M)^T, \text{since}(A^{-1})^T = (A^T)^{-1} \\
(G^{-1}M)^{-T} &= (G^{-1}M) \\
&\Rightarrow G^{-1}M \text{ diagonal since } G^{-1}M \text{ lower triangular and } (G^{-1}M)^{-T} \text{ upper triangular} \\
&\Rightarrow G^{-1}M = D \Rightarrow M = GD \\
I &= (G^{-1}GD)(G^{-1}GD)^T \\
I &= DD^T = D^2 \Rightarrow \text{ so the entries of D are on the order of 1}
\end{aligned}$$

4.4 Schur complement

Observe A can be written in the following form $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$. If we run the LU factorization algorithm for k steps, the resulting $A' = A$ is equal to $A = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \begin{bmatrix} I & A_{21}A_{11}^{-1} \\ 0 & I \end{bmatrix}$
The bottom-right block of $A' = A, A'_{22} = A_{22}$ is equal to $A_{22} - A_{21}A_{11}^{-1}A_{12}$ from the original matrix. This is called the **Schur complement** of A

4.4.1 Schur complement derivation

At any step in the LU factorization, A can be written in the form $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$
From this equality, we can create a system of equations and derive $A_{22} - A_{21}A_{11}^{-1}A_{12} = L_{22}U_{22}$. Last, show that A'_{22} in the LU factorization is equal to $A_{22} - L_{21}U_{12}$ since at each step we're subtracting $l_iU_i^T$, which can be stored as the nonzero rows/columns of $L_{21}U_{12}$.

5 QR factorization

The QR factorization decomposes a matrix, $A \in \mathbb{R}^{m\times n}, m \geq n$ into an orthogonal (orthonormal) matrix, Q and an upper triangular matrix, R. Recall $Q \in \mathbb{R}$, orthogonal, $Q^TQ = I$. QR factorization can take two forms when A skinny. $Q \in \mathbb{R}^{m\times m}$ can be square and $R \in \mathbb{R}^{m\times n}$ can be skinny. Or $Q \in \mathbb{R}^{m\times n}$ can be skinny and $R \in \mathbb{R}^{n\times n}$ can be square.

5.1 The QR factorization is unique

Proof that the QR factorization is unique for full rank matrix, A:

$$A = QR \Rightarrow Q^TA = R \Rightarrow R^TQ^TA = R^TR \Rightarrow (QR)^TA = R^TR \Rightarrow A^TA = R^TR$$

So A^TA can be written as R^TR, which is the structure of the Cholesky factorization. Suffice to show that A^TA is Symmetric and Positive Definite.

$$\begin{aligned} \text{Symmetric: } & (A^TA)^T = A^TA \\ \text{Positive definite: for } & x \neq 0, \\ & x^TA^TAx = (Ax)^T(Ax) = (QRx)^T(QRx) = x^TR^TQ^TQRx = (Rx)^T(Rx) \\ & \text{Rx is of the form } Rx = \begin{bmatrix} r_{11}x_1 \\ r_{12}x_1 + r_{22}x_2 \\ \vdots \\ \sum_{i=1}^n r_{in}x_i \end{bmatrix}, \text{ so } (Rx)^T(Rx) = \sum_{i=1}^n (\sum_{j\leq i} r_{ij}x_j)^2 \\ & \text{So, } (Rx)^T(Rx) > 0 \text{ for } x \neq 0 \end{aligned}$$

5.2 Householder reflection

Construct Q^T for each column in A that projects it onto a corresponding column of R. Start with a_1 and find Q_1^T such that $Q_1^Ta_1 = r_1$, where $r_1 = \pm \|a_1\|_2 e_1$ (since Q^T is orthogonal)
The key to the iterative part of the algorithm is to construct $Q_i^T, i > 1$ with an identity matrix in the upper-left $i-1 \times i-1$ quadrant, and a smaller Q_i^{*T} in the lower right $n-i \times n-i$ quadrant, filling the remaining sections of the matrix with 0's

5.2.1 Constructing the Householder reflection permutation

The **Householder reflection** maps $a \to \|a\|_2 e_1$ with $P = I - \beta vv^T$, where $v = a - \|a\|_2 e_1$, and $\beta = 2/v^Tv$

- Multiplying Px is the same as taking the vector x and subtracting $\frac{2vv^T}{v^Tv}x$ from it, where $\frac{2vv^T}{v^Tv}x$ is twice the projection of x onto v.
- In householder reflection, $a + \|a\|_2 e_1$ is the line of reflection. Perpendicular to this line is $a - \|a\|_2 e_1$, the line of reflection

Aside: The fixed points of a reflection, P, are the points that remain unchanged when multiplied by the reflection, $Px = x$. Geometrically, these are the points that are *orthogonal* to the vector v defining the reflection (i.e., $v^Tx = 0$)

5.3 Givens transformation

The Givens transformation is a precise, higher-cost QR factorization. A **Givens rotation** rotates $u = (u_1, u_2)^T$ to $\|u\|_2 e_1$. The matrix that does this, G^T, is defined by $G^T = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, c = \frac{u_1}{\|u\|_2}, s = -\frac{u_2}{\|u\|_2}$.
A full matrix, P_i, can be constructed to only contain this targeted transformation. Sequentially, the P_i's can multiply A to arrive at R

5.4 Gram-Schmidt transformation

When $A \in \mathbb{R}^{m \times n}$ is tall and thin. The **Gram-Schmidt Transformation** starts with the property that $A = QR$ can be written as a sum of the outer products of the columns of Q and rows of R: $A = QR = q_1 r_1^T + \dots q_m r_m^T$.

$$
\begin{aligned}
r_{11} &= \|a_1\|_2 \text{, since } \|a_1\|_2 = \|q_1 r_{11}\|_2 \text{ and } q_i \text{ orthogonal} \\
q_1 &= \frac{1}{r_{11}} a_1 \text{, since } a_1 = q_1 r_{11} \text{ by construction of } QR \\
r_{1j} &= q_1^T a_j \text{, (repeat for all } j\text{) since} \\
(a_j &= q_1 r_{1j} + \cdots + q_j r_{jj}) \\
(q_1^T a_j &= q_1^T q_1 r_{1j} + \cdots + q_1^T q_j r_{jj}) \\
(q_1^T a_j &= r_{1j}) \text{, since } q_i \text{ orthonormal} \\
A' &= A - q_1 r_1^T \text{Repeat for } A'
\end{aligned}
$$

5.5 QR factorization to solve least-squares problems

When A is tall and thin, unlikely that we get a solution to $Ax = b$. Instead, we solve $argmin_x \|Ax - b\|_2$.

5.5.1 Method of normal equations

Assuming A full rank. x which solves $argmin_x \|Ax - b\|_2$ when $b - Ax$ is orthogonal to the range of A.

$$
\begin{aligned}
\text{Want: } (b - Ax) &\perp \{z | z = Ay\} \\
(b - Ax) &\perp range(A) \Rightarrow (b - Ax) \perp a_i, \forall i \in A \\
a_1^T (b - Ax) &= 0, \forall i \in A \Rightarrow A^T (b - Ax) = 0 \Rightarrow x = (A^T A)^{-1} A^T b
\end{aligned}
$$

Method can run into issues when A is poorly conditioned. Notice, condition number of $A^T A, \kappa(A^T A) = \kappa(A)^2$.

5.5.2 QR method for least squares

Assuming A full rank. The QR method for least squares attempts to address the issue of poor conditioning.

$$
\begin{aligned}
A^T (Ax - b) = 0 &\Rightarrow R^T Q^T (Ax - b) = 0 \\
Q^T (Ax - b) &= 0 \text{, since we assume } A, R \text{ full rank (multiply both sides by } R^{-T}) \\
Q^T Q R x - Q^T b = 0 \Rightarrow Rx &= Q^T b \Rightarrow x = R^{-1} Q^T b
\end{aligned}
$$

5.5.3 SVD for rank-deficient A

When A not full rank. Add the additional criteria $\min_x \|x\|_2$. Solve using thin SVD.

$$
\begin{aligned}
(Ax - b) &\perp range(U) \text{, since } R(A) = R(U) \text{ for } A = U \Sigma V^T \\
U^T (Ax - b) = 0 &\Rightarrow U^T (U \Sigma V^T x - b) = 0 \Rightarrow \Sigma V^T x = U^T b \\
x &= V \Sigma^{-1} U^T b
\end{aligned}
$$

$\min_x \|x\|_2$ the $x \perp N(A)$, the shortest vector between $N(A)$ and the vector/plane of solutions to $argmin_x \|Ax - b\|_2$. This value it turns out must be in $R(V)$ since $R(V) = N(A)^\perp$