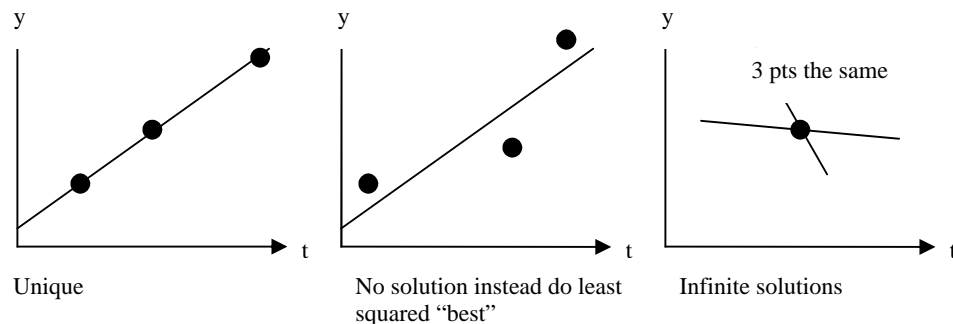


CS205 - Class 3

1. So far we have discussed solving $Ax=b$ for square $n \times n$ matrices A . For more general $m \times n$ matrices, there are a variety of scenarios.
 - a. When $m < n$, the problem is **underdetermined** since there is not enough information to determine a unique solution for all the variables. Usually $m < n$ implies that there are *infinite solutions*. However, in some cases, there may be contradictory equations leading to the absence of any solutions.
 - b. When $m > n$, the problem is **overdetermined** although this in itself doesn't tell us everything about the nature of the solution. For example, if enough equations are linear combinations of each other, there can still be a unique solution or infinite solutions.
 - i. We can use the rank of the matrix to enumerate the possibilities. Recall that the **rank** of a matrix is the number of linearly independent columns that it has. Thus a $m \times n$ matrix has at most a rank of n .
 - ii. If the rank $< n$ some columns are linear combinations of others and we say that the matrix is **rank-deficient** and there may be an infinite number of solutions.
 - iii. On the other hand, if the rank $= n$, i.e. all the columns are linearly independent and we are guaranteed either a unique solution or no solution. In the case of no solution, there is the notion of the "closest fit" in a **least squares** sense. That is, the least squares solution finds the closest possible solution.
2. When solving systems of equations, we want $Ax=b$. We define the **residual** as $r=b-Ax$, and note that the residual gives us some measure of the error. Of course the goal is to attain $r=0$.
3. The **least squares method** finds the *best-fit* or best approximation to the solution in the sense that the least squares solution x minimizes the L_2 norm of the residual, i.e. it minimizes $\|r\|_2$.
4. An example, consider interpolating a given a set of m data points, (t_i, y_i) , with a straight line $y = x_1 + x_2 t$. Here, each data point leads to a new equation, for example with three data points ($m=3$) we obtain

$$\begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$



- a. Here, if one gave the exact same pair (t_i, y_i) three times, then there is really only one point and there are an infinite number of lines that go through it.
- b. If the three points all line on the same line, then that line is a unique solution to the problem.
- c. If the three points do not all lie on a line, the problem is overdetermined and there is no solution, i.e. no straight line that passes through the points. In this case, we can look for the best least squares solution, or the line that passes as close as possible to the three points.

- d. Often when doing computer vision or graphics applications you can collect more data to make your matrix overdetermined. You might find that some variables have no effect as well and you might decide to drop them.
5. One method of solving both least squared (and unique) problems are the **normal equations**. If the columns of A are linearly independent, $A^T A x = A^T b$ has a unique solution.
- $A^T A$ is always square as $(m \times n)(n \times m) = (m \times m)$ and symmetric and positive definite. Thus we can use Cholesky factorization i.e. $A^T A = LL^T$ or $LL^T x = A^T b$.
 - How do we know that the matrix $A^T A$ is symmetric and positive definite? The symmetry is immediate from $(A^T A)^T = A^T A^{TT} = A^T A$. There are important classes of matrices that come up often in practice that have the form of $A^T A$. Consider any non-zero vector x . Then $x^T A^T A x = y^T y \geq 0$ with equality only when $y = Ax$ is zero, which can only occur if A does NOT have full rank.
 - This sounds too good to be true, and it is. The condition number of $A^T A$ is equal to the condition number of A squared, which can lead to real problems if A is already poorly conditioned.
6. **QR factorization** $A = QR$ where Q is a $m \times n$ orthogonal matrix with orthonormal columns and R is a $n \times n$ upper triangular matrix.
- $Q^T Q = I$ since columns of Q are orthonormal. Likewise $Q Q^T = I$.
 - We transform $Ax = b$ into $QRx = b$ and then applying Q^T to each side of the equation leads to $Rx = Q^T b$ which can be solved with an upper triangular solve.
 - Solving $Ax = b$ with a QR factorization results in the least squares solution to the problem. To show this, we know that the normal equations system $A^T A x = A^T b$ gives the least squares solution as $x = (A^T A)^{-1} A^T b$. If we let $A = QR$ then we get $A^T = (QR)^T = R^T Q^T$ so
 $A^T A = (R^T Q^T)(QR) = R^T (Q^T Q)R = R^T I R = R^T R$ and finally
 $x = (A^T A)^{-1} A^T b = (R^T R)^{-1} R^T Q^T b = R^{-1} R^{-T} R^T Q^T b = R^{-1} Q^T b$ which is $Rx = Q^T b$.
 - Therefore, the least squares solution for the initial problem $Ax = b$ coincides with that of the transformed system $Rx = Q^T b$ (which is unique provided that A is not rank deficient). Note that using Gaussian elimination on A does not preserve the least squares solution.
7. The **modified Gram-Schmidt** method orthonormalizes the columns of a matrix A resulting in both Q and R.
- Algorithm
 - for $k=1, n$ - for each column
 - $r_{kk} = \|\vec{a}_k\|_2$, $\vec{a}_k = \vec{a}_k / r_{kk}$ ($= \vec{q}_k$) - rescale the column to unit length
 - for $j=k+1, n$ - for all the rest of the columns to the right of column k
 - $r_{kj} = \vec{a}_k \cdot \vec{a}_j$ ($= \vec{q}_k \cdot \vec{a}_j$), $\vec{a}_j = \vec{a}_j - r_{kj} \vec{a}_k$ ($= r_{kj} \vec{q}_k$) - orthogonalize each column with \vec{a}_k (which is actually \vec{q}_k)
 - Notice that the way the above loop is written, the A matrix can be factored “in place”. That is, the elements of matrix A are replaced with the elements of matrix Q as this process is carried out. This is important for the very large matrices that often arise in practice.
 - It is often useful to think about this procedure with matrices and to think about these matrices geometrically. We define a projection matrix P as one which has the property that $P^2 = P$. Intuitively, what this means is that once you project onto a subspace, projecting there again does nothing (you’re already there).
 - Consider the matrix $I - \vec{q}_k \vec{q}_k^T$. It is easy to verify the fact that it is a projection matrix using the fact that \vec{q}_k is a unit vector. It has the effect of zeroing out the component of a vector that is in the

direction of \vec{q}_k . The idea of Gram-Schmidt is to project to force orthogonality and then normalize to force normality.

e. The resulting \vec{q}_k are the columns of Q and the r_{kj} with $j \geq k$ are the entries of the upper triangular R.

f. Consider $A = \begin{bmatrix} 3 & -3 & 3 \\ 2 & -1 & 1 \\ 2 & -3 & 3 \\ 2 & -3 & 5 \end{bmatrix}$.

i. Here $r_{11} = \|\vec{a}_1\| = 5$ and we rescale the first row to unit length as $\vec{q}_1 = \vec{a}_1 / r_{11} = \begin{bmatrix} .6 \\ .4 \\ .4 \\ .4 \end{bmatrix}$.

ii. Then we compute the overlap between this column and the other two columns as $r_{12} = \vec{q}_1 \cdot \vec{a}_2 = -5$ and $r_{13} = \vec{q}_1 \cdot \vec{a}_3 = 5$. Note that we now have the first (top) row of R. Next we

subtract out the overlap with the first column using $\vec{a}_j - r_{kj}\vec{q}_k$ to obtain $\begin{bmatrix} .6 & 0 & 0 \\ .4 & 1 & -1 \\ .4 & 1 & -3 \\ .4 & -1 & 1 \\ .4 & -1 & 3 \end{bmatrix}$.

iii. We then move to the second column where $r_{22} = \|\vec{a}_2\| = 2$ and we rescale the second row to unit

length $\vec{q}_2 = \vec{a}_2 / r_{22} = \begin{bmatrix} 0 \\ .5 \\ .5 \\ -.5 \\ -.5 \end{bmatrix}$.

iv. Then we compute the overlap between this column and the remaining third column as

$r_{23} = \vec{q}_2 \cdot \vec{a}_3 = -4$. Then we subtract out the overlap using $\vec{a}_j - r_{kj}\vec{q}_k$ to obtain $\begin{bmatrix} .6 & 0 & 0 \\ .4 & .5 & 1 \\ .4 & .5 & -1 \\ .4 & -.5 & -1 \\ .4 & -.5 & 1 \end{bmatrix}$.

v. We then move to the third column where $r_{33} = \|\vec{a}_3\| = 2$ and we rescale the third row to unit

$$\text{length } \vec{q}_3 = \vec{a}_3 / r_{33} = \begin{bmatrix} 0 \\ .5 \\ -.5 \\ -.5 \\ .5 \end{bmatrix} \text{ and our final Q matrix is } \begin{bmatrix} .6 & 0 & 0 \\ .4 & .5 & .5 \\ .4 & .5 & -.5 \\ .4 & -.5 & -.5 \\ .4 & -.5 & .5 \end{bmatrix} \text{ and } R = \begin{bmatrix} 5 & -5 & 5 \\ 0 & 2 & -4 \\ 0 & 0 & 2 \end{bmatrix}.$$

8. Transition to Householder. So far, we've seen two different ways of finding a least squares solution, namely the normal equations and the (modified) Gram-Schmidt method. Using the normal equations squares the condition number of the A matrix, which could be bad to begin with. The Gram-Schmidt method does not suffer from the same numerical problems, but can still be unstable. If the columns of A are "nearly linearly-dependent," then this method can suffer from numerical instabilities. The QR decomposition is exactly what the Gram-Schmidt procedure does couched in the language of matrix factorization. There is an algorithm to performs the QR factorization that does not suffer from the numerical instabilities mentioned above and this is the one that is quite often used by practitioners.

9. A **Householder transform** is defined by $H = I - \frac{2}{v^T v} v v^T$ for some vector $v \neq 0$. Note that $H = H^T = H^{-1}$, and thus H is orthogonal.

a. For a vector a , we define H_k using $v_k = \hat{a} + S(a_k) \|\hat{a}\|_2 e_k$ where $\hat{a} = (0, \dots, 0, a_k, \dots, a_m)^T$ and $S(a_k) = \pm 1$ is the sign function. Then $H_k a$ zeroes out the entries of a below a_k .

b. Let $a = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$. Then $v_1 = \hat{a} + S(a_1) \|\hat{a}\|_2 e_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} + S(2) \sqrt{9} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$.

c. Note that $Ha = a - \frac{2v^T a}{v^T v} v$ so we never need to form H explicitly, but instead only need to find the vector v .

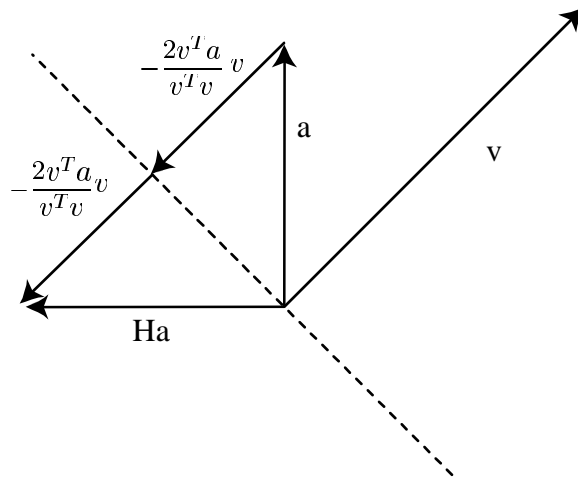
d. $H_1 \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \frac{2(5,1,2)(2,1,2)^T}{(5,1,2)(5,1,2)^T} \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \frac{2 \times 15}{30} \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$.

e. In order to use the householder transform to compute the QR factorization of a matrix A, we define H_k using $v_k = \hat{a}^{(k)} + S(a_{kk}) \|\hat{a}^{(k)}\|_2 e_k$ where $\hat{a}^{(k)} = (0, \dots, 0, a_{kk}, \dots, a_{mk})^T$. When applying H_k to the matrix A, $H_k A$ is obtained by applying H_k to every column of A.

f. Overall, we obtain $H_n \cdots H_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$. Then applying $Q = H_1^T \cdots H_n^T$ (all the H's are orthogonal and so

is their product) to both sides of the equation results in $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$.

- g. Thus, to solve $Ax=b$, we write $Q \begin{bmatrix} R \\ 0 \end{bmatrix} x = b$, and $\begin{bmatrix} R \\ 0 \end{bmatrix} x = Q^T b = H_n \cdots H_1 b = \hat{b}$, and then solve $\begin{bmatrix} R \\ 0 \end{bmatrix} x = \hat{b}$ with an upper triangular solve. That is, starting with $Ax=b$, we simply apply $Q^T = H_n \cdots H_1$ to each side, one H_k at a time to obtain $\begin{bmatrix} R \\ 0 \end{bmatrix} x = \hat{b}$, and then solve this upper triangular problem.
- h. Again, thinking geometrically, we can see that the Householder transformations are actually reflections about the hyperplane orthogonal to v . This can be seen by H subtracting off twice the component of x in the direction of v .



- i. We motivated this entire QR discussion by solving an overdetermined system of linear equations. Often there will be no solution. There are too many constraints, but we can still strive for the best solution, the least-squares solution. We have shown that factorization preserves the least squares solution while Gaussian elimination fails to do so. The intuition behind this fact is that the matrix Q shuffles the equations in such a way that the least squares solution exactly solves the first n equations, while leaving the minimum amount of slack in the remaining $m-n$ equations. Gaussian elimination fails to shuffle the equations in a way that keeps the norm of the residual at a minimum.