

# STATS 305A - Lecture 2

John Duchi  
Notes by Michael Howes

September 26, 2021

## 1 Outline

Today we will do two things. A matrix and linear algebra overview. Covering

- (a) Independence,
- (b) Rank/ invertibility/ solving  $Ax = b$ ,
- (c) Decompositions.

We will also discuss some basic of optimisation. Detailed linear algebra notes will be put on the course webpage.

## 2 Linear algebra review

### 2.1 Vectors

Vector are  $x \in \mathbb{R}^n$ , we will write

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad x_i \in \mathbb{R}.$$

We will not use row vectors. All vectors are column vectors. The *norm* of a vector is its size. We will must commonly use the Euclidean or 2-norm. That is we will define

$$\|x\| = \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{x^T x}.$$

We will occasionally use  $p$ -norms for  $p \in [1, \infty]$  given by

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}.$$

When  $p = 1$ , we have  $\|x\|_1 = \sum_{i=1}^n |x_i|$ . When  $p = 2$ ,  $\|x\|_2$  is the Euclidean norm and when  $p = \infty$ ,  $\|x\|_\infty = \max\{|x_i| : i = 1, \dots, n\}$ . These are the values of  $p$  we will most commonly use.

We will say that  $k$ -vectors  $x_1, \dots, x_k$  are (*linearly*) *independent* if for all  $\alpha \in \mathbb{R}^k$ ,

$$\sum_{i=1}^k \alpha_i x_i = 0 \iff \alpha_i = 0 \text{ for } i = 1, 2, \dots, k.$$

This is equivalent to requiring that for all  $\alpha, \eta \in \mathbb{R}^k$ ,

$$\sum_{i=1}^k \alpha_i x_i = \sum_{i=1}^k \beta_i x_i \iff \alpha_i = \beta_i \text{ for } i = 1, 2, \dots, k.$$

The *linear span* of the collection of vectors  $\{x_i\}_{i=1}^k$  is the set

$$\text{span}\{x_i\} = \left\{ \sum_{i=1}^k \alpha_i x_i : \alpha \in \mathbb{R}^k \right\}.$$

Note that a collection of vectors  $\{x_i\}_{i=1}^k$  are independent if and only if for  $i = 1, \dots, k$ ,  $x_i$  is not in  $\text{span}\{x_j\}_{j \neq i}$ .

## 2.2 Matrices

We will now look at solving and understanding the matrix equation  $Ax = b$ . This does not always have a solution. Suppose  $A \in \mathbb{R}^{m \times n}$ , then we have

$$\begin{aligned} A &= [a_1, \dots, a_n], \quad a_i \in \mathbb{R}^m, \\ &= \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_m^T \end{bmatrix}, \quad \tilde{a}_j \in \mathbb{R}^n. \end{aligned}$$

The vectors  $a_i$  are the columns of  $A$  and the transposed vectors  $\tilde{a}_j^T$  are the rows of  $A$ . The *rank* of  $A$  is

$$\# \text{ of independent columns of } A = \# \text{ of independent rows of } A.$$

The rank of a matrix is a very unstable quality. We may have rank 0, then add a small amount of noise and suddenly have rank  $n$ . This makes it not very useful in statistics but we will use it occasionally.

Given  $A \in \mathbb{R}^{m \times n}$ , we say that  $A$  has a *left inverse*  $B \in \mathbb{R}^{n \times m}$  such that

$$BA = I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

The matrix  $I_n$  is called the  $n \times n$  identity. We say that  $A$  has a right inverse  $C \in \mathbb{R}^{n \times m}$  if  $AC = I_m$ . If  $A$  has a left inverse  $B$  and a right inverse  $C$ , then  $B = C$  and we define  $A^{-1} := B = C$ . In this case we say that  $A$  is invertible with inverse  $A^{-1}$ . To see why  $B = C$ , note that

$$B = BI_m = B(AC) = (BA)C = I_n C = C.$$

**Definition 1.** The *range* or *column space* of a matrix  $A$  is equal to

$$\text{span}\{a_i\}_{i=1}^n = \{Ax : x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

We have equality in the above line since  $Ax = \sum_{i=1}^n a_i x_i$ .

**Definition 2.** The *null space* of  $A$  is

$$\text{null}(A) = \{x \in \mathbb{R}^n : Ax = 0\} \subseteq \mathbb{R}^n.$$

The null space will be important in our linear models. If  $Y = X\beta + \varepsilon$ , then  $\text{null}(X)$  contains all the directions in  $\beta$  that we get no information about. Even if  $\text{null}(X) = \{0\}$ , there may be problems if our matrix is “ill-conditioned”. This means that there are vectors that are “close to”  $\text{null}(X)$ . These are directions of  $\beta$  in which it is hard (but not impossible) to get information.

**Theorem 1.** [The rank-nullity theorem] *Let  $A \in \mathbb{R}^{n \times m}$ , then  $\text{null}(A) = \text{range}(A^T)^\perp$ , where for  $S \subseteq \mathbb{R}^n$ ,  $S^\perp = \{y \in \mathbb{R}^n : y^T x = 0, \text{ for all } x \in S\}$ .*

*Proof. (sketch)* If  $x \in \text{null}(A)$ , then  $Ax = 0$ . Thus for  $y = A^T w \in \text{range}(A^T)$  we have

$$y^T x = w^T (A^T)^T x = w^T (Ax) = w^T 0 = 0.$$

Thus  $y \in \text{range}(A^T)^\perp$ . If  $x \in \text{range}(A^T)^\perp$ , then for all  $w \in \mathbb{R}^n$ ,

$$0 = (A^T w)^T x = w^T (Ax),$$

which implies  $Ax = 0$ . That is  $x \in \text{null}(A)$ . □

### 2.3 Special matrices and matrix decompositions

**Definition 3.** A matrix  $Q \in \mathbb{R}^{n \times n}$  is *orthogonal* if  $Q^T Q = Q Q^T = I_n$ . If  $Q = [q_1, q_2, \dots, q_n]$ , then this is equivalent to

$$q_i^T q_j = \begin{cases} 1 & \text{if } i = j, \\ 0^{\text{else.}} \end{cases}$$

That is the vectors  $\{q_1, \dots, q_n\}$  are an orthonormal set. [Aside: if  $Q \in \mathbb{R}^{n \times n}$  and  $Q^* Q = I$ , then  $Q$  is said to be orthonormal or unitary].

If  $U \in \mathbb{R}^{n \times m}$  where  $m \geq n$  (more rows than columns, tall matrix), then  $U$  will also be called orthogonal if  $U U^T = I_n$ . Note that if  $m > n$ , then  $U^T U$  cannot equal  $I_m$ . Again this is equivalent to the columns of  $U$  being orthogonal.

We now will examine a number of matrix factorisations. A lot of computational matrix calculations/statistics/ML involves reducing a matrix  $A$  to a product of simpler matrices. In particular we can use these to solve the equation  $Ax = b$ .

**Definition 4.** A matrix  $R \in \mathbb{R}^{n \times n}$  is *upper (or right) triangular* if  $R$  is of the form

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ 0 & r_{2,2} & \dots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{n,n} \end{bmatrix}.$$

That is  $R$  has all zeros below the main diagonal.

If we are asked to solve  $Rx = b$  and  $R$  is upper triangular we can use the following “back-substitution” algorithm. First write the equation as

$$\begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n-1} & r_{1,n} \\ 0 & r_{2,2} & \dots & r_{2,n-1} & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & r_{n-1,n-1} & r_{n-1,n} \\ 0 & 0 & \dots & 0 & r_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

Assume that  $r_{i,i} \neq 0$  for all  $i = 1, \dots, n$ . Thus we can see that we must have  $r_{n,n} x_n = b_n$ . Thus  $x_n = \frac{b_n}{r_{n,n}}$ . We can then move up to the next row where we have

$$\begin{aligned} r_{n-1,n-1} x_{n-1} + r_{n-1,n} x_n &= b_{n-1} \\ \therefore x_{n-1} &= \frac{1}{r_{n-1,n-1}} (b_{n-1} - r_{n-1,n} x_n) \\ &= \frac{1}{r_{n-1,n-1}} \left( b_{n-1} - r_{n-1,n} \frac{b_n}{r_{n,n}} \right), \end{aligned}$$

since we previously showed  $x_n = \frac{b_n}{r_{n,n}}$ . Continuing in this way we can see that if we have solved for  $x_n, x_{n-1}, \dots, x_{k-1}$  we can solve for  $x_k$ . This takes approximately  $n^2$  operations to perform which is pretty much optimal.

What do we do if we have a matrix  $A$  that isn't upper triangular? We can use the QR factorisation.

**Theorem 2.** *Let  $A \in \mathbb{R}^{m \times n}$  be a matrix with  $m \geq n$  ( $A$  is a tall matrix), then there exist  $Q \in \mathbb{R}^{m \times n}$  and  $R \in \mathbb{R}^{n \times n}$  such that  $R$  is upper triangular and invertible,  $Q^T Q = I_n$  and  $A = QR$ . The matrices  $Q$  and  $R$  are called the QR factorisation of  $A$ .*

Note that if we have the QR factorisation of  $A$ , then we can easily solve  $Ax = b$ . Since this is because  $Ax = b$  implies  $Q^T Ax = Q^T b$ . Since  $A = QR$  we have  $Q^T A = Q^T QR = R$  and thus the solution to  $Rx = Q^T b$  solves the original equation  $Ax = b$ . The QR factorisation can be constructed iteratively via the Gram-Schmidt algorithm.

Given  $A = [a_1, a_2, \dots, a_n] \in \mathbb{R}^{m \times n}$  and  $k \leq n$  we will produce

$$Q_k = [q_1, \dots, q_k] \in \mathbb{R}^{m \times k} \text{ and } R_k \in \mathbb{R}^{k \times k},$$

such that

- $R_k$  is upper triangular,
- $\text{span}\{q_i\}_{i=1}^k = \text{span}\{a_i\}_{i=1}^k$ ,
- $Q_k^T Q_k = I_k$ , and
- $Q_k R_k = [a_1, a_2, \dots, a_k]$ .

When we reach  $k = n$  we will be done. We will show how to do this for  $k = 1, 2$  and generalise.

Start  $k = 1$ : Define  $q_1 = \frac{a_1}{\|a_1\|}$ ,  $r_{11} = \|a_1\|$ , then  $r_{11}q_1 = a_1$  and  $q_1^T q_1 = \frac{a_1^T a_1}{\|a_1\|^2} = 1$ .

Next step  $k = 2$ : We already have  $q_1$  and we want  $q_2$ . We want to preserve the span and so  $q_2$  should be a linear combination of  $a_2$  and  $q_1$ . We also want  $q_2$  to be orthogonal to  $q_1$ . Thus we subtract off the  $q_1$  part of  $a$  and define

$$v = a_2 - a_2^T q_1 q_1.$$

Then  $v^T q_1 = a_2^T q_1 - a_2^T q_1 q_1^T q_1 = 0$ . We want  $q_2$  to have norm 1 and thus define  $q_2 = \frac{v}{\|v\|}$ . Note that the equation  $a_2 = \|v\| q_2 + a_2^T q_1 q_1$  shows that  $r_{2,2} = \|v\|$  and  $r_{1,2} = a_2^T q_1$ . Thus we have constructed  $R_2$  as well.

Inductive case: Suppose that we have just finished the  $k - 1$  step. Generalising what we did before we can set

$$v = q_k - \sum_{i=1}^{k-1} a_k^T q_i q_i.$$

Then  $v^T q_i = 0$  for  $i = 1, \dots, k - 1$  and we can define  $q_k = \frac{v}{\|v\|}$ . Since again  $a_k = \|v\| q_k + \sum_{i=1}^{k-1} a_k^T q_i q_i$ , we have that

$$r_{i,k} = \begin{cases} a_k^T q_i & \text{if } i < k, \\ \|v\| & \text{if } i = k. \end{cases}$$

Thus we have constructed  $R$  as well. Note that at each step we add a new column of  $R$ .

The QR algorithm is implemented on any scientific programming language you might use (R, python, julia, etc). You do not need to use Gram-Schmidt by hand. Indeed Gram-Schmidt is rarely used because it is numerically unstable and so other methods are used. Gram-Schmidt is good for the theoretical justification that the QR decomposition exists.

## 2.4 Eigen-decompositions

Let  $A \in \mathbb{R}^{n \times n}$ , a vector  $v$  is an eigenvector with eigenvalue  $\lambda$  if  $v \neq 0$  and  $Av = \lambda v$ . Most of our analysis will involve symmetric matrices (those that satisfy  $A^T = A$ ). This simplifies the analysis of eigenvalues a lot.

**Theorem 3.** [The Spectral Theorem] *If  $A$  is a symmetric matrix, then  $A$  has an orthonormal set of eigenvalues  $v_1, \dots, v_n$  with real eigenvalue  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ . That is*

$$Av_i = \lambda_i v_i \text{ for } i = 1, \dots, n,$$

and for  $i, j = 1, \dots, n$ ,

$$v_i^T v_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

If we set  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  and  $V = [v_1, v_2, \dots, v_n]$ , then the spectral theorem states that  $A = V\Lambda V^T$  and  $V^T V = I_n$ . To see why this holds, note that  $A = V\Lambda V^T$  if and only if  $AV = V\Lambda$ . By matrix multiplication  $AV = [Av_1, \dots, Av_n]$  and  $V\Lambda = [\lambda_1 v_1, \dots, \lambda_n v_n]$ . Thus we have  $AV = V\Lambda$  if and only if  $Av_i = \lambda_i v_i$  for each  $i$ . We also previously saw that the vector  $\{v_i\}_{i=1}^n$  are orthonormal if and only if  $V^T V = I_n$ .

A proof of the spectral theorem will be given in the document that is to be put on the course webpage. The spectral theorem makes calculations easier since  $A^k = V\Lambda^k V^T$  for  $k = 1, 2, \dots$  and if  $A$  is invertible  $A^{-k} = V\Lambda^{-k} V^T$ .

## 2.5 Singular value decomposition

The spectral theorem assumes that  $A$  is symmetric. The following decomposition works for all matrices. Each matrix  $A \in \mathbb{R}^{m \times n}$  with  $m \geq n$  ( $A$  is tall) has an SVD (singular value decomposition) into

$$A = U\Sigma V^T,$$

where  $U \in \mathbb{R}^{m \times n}$ ,  $\Sigma \in \mathbb{R}^{n \times n}$  is diagonal with diagonal entries  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ ,  $V \in \mathbb{R}^{n \times n}$  and  $U^T U = I_n = V^T V$ . This decomposition tells us how  $A$  acts on any vector  $x \in \mathbb{R}^n$ .

Given  $x \in \mathbb{R}^n$ , we first change it to the basis of  $V$ , then we multiply by gains  $\sigma_1, \dots, \sigma_n$  and then give the output in terms of the  $U$  basis. This is what the equation  $Ax = U\Sigma V^T x$  tells us.

Note that  $A^{-1} = V\Sigma^{-1}U^T$  if  $\sigma_i > 0$  for all  $i$  and  $m = n$ . Again see notes for the construction of the SVD.

## 3 Optimisation

We didn't have time to cover this today.