INF1004 L2 Linear and Affine Equation Systems

Alexander Binder

SIT

February 25, 2023

Inner Product

- is the product of vector norms times the cosine of the angle between the two vectors
- a similarity measure

Inner product

Let be $u \in \mathbb{R}^d$, $v \in \mathbb{R}^d$, Definition: Then the inner product between u and v is defined as:

$$u \cdot v = \sum_{k=1}^{d} u_k v_k$$

$$[1, -3, 5, 2] \cdot [2, 3, 4, -1] = 1 * 2 + (-3 * 3) + 5 * 4 + 2 * (-1) = 11$$

Interpretation of the inner product

It holds for the inner product defined above that:

$$u \cdot v = \|u\|_2 \|v\|_2 \cos(\angle(u, v))$$

it is the product of the euclidean length of u, of v and the cosine of the angle between these two vectors.

The canonical inner product defines the euclidean norm via:

$$\|v\|_2 = \sqrt{v \cdot v}$$

$$\|[1, -3, 5, 2]\|_2 = \sqrt{1 + (-3)^2 + 5^2 + 2^2} = \sqrt{39}$$

Norms

- A way to measure length of vectors
- Euclidean norm is well known.

$$\|v\|_2 = \left(\sum_{k=1}^{d} v_k^2\right)^{1/2}$$

Other norms are in use, as well:

$$\|v\|_p = \left(\sum_{k=1}^{d} |v_k|^p\right)^{1/p}$$

Difference to the Euclidean norm: Replace 2 by p, use $|v_k|$ for a vector component

Norms

Properties of any norm

- $\|0\| = 0$
- $\|\lambda v\| = |\lambda| \|v\|$
- $\|v + u\| \leq \|v\| + \|u\|$
- a norm induces a distance measure $d(\cdot, \cdot)$ via $d(u, v) = \|u - v\|$

Projecting onto a vector and removing the direction of a vector

Project a vector x onto a vector v:

$$x_{\parallel v} = \frac{x \cdot v}{v \cdot v} v = \left(x \cdot \frac{v}{\|v\|_2} \right) \frac{v}{\|v\|_2}$$

Remove from a vector x a vector v:

$$x_{\perp v} = x - x_{\parallel v}$$

Definition of a vector space

The vector space spanned by a set of vectors $w^{\{0\}}, \ldots, w^{\{k-1\}}$ is the set of all their linear combinations using all possible real numbers $a_0, \ldots, a_{k-1} \in \mathbb{R}$

$$V = \{w \text{ such that}$$
$$w = \sum_{r=0}^{k} a_r w^{\{r\}}, a_0, \ldots, a_{k-1} \in \mathbb{R}\}$$

The dimension of this set is the largest number of independent vectors which we can obtain from $w^{\{0\}}, \ldots, w^{\{k-1\}}$.

Matrix-vector multiplication

Given a matrix of shape (n, d) and a vector of dimensionality d, the multiplication Ax of them (with x being on the right hand side) is defined as a vector of length n such that

$$Ax = \begin{bmatrix} a_{0,0} & \cdots & a_{0,d-1} \\ a_{1,0} & \cdots & a_{1,d-1} \\ \vdots & \ddots & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,d-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_{d-1} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} a_{0,0} & \cdots & a_{0,d-1} \end{bmatrix} \cdot x \\ \begin{bmatrix} a_{1,0} & \cdots & a_{1,d-1} \end{bmatrix} \cdot x \\ \vdots \\ \begin{bmatrix} a_{n-1,0} & \cdots & a_{n-1,d-1} \end{bmatrix} \cdot x \end{bmatrix}$$

Thus, Ax is a vector and the k-th component of vector Ax is given as an inner product

$$\begin{aligned}(Ax)_k &= A[k,:] \cdot x \\ &= \left(a_{k,0} \dots a_{k,d-1}\right) \cdot x \\ &= \sum_{r=0}^{d-1} a_{k,r} x_r\end{aligned}$$

between the k-th row of A and vector x. The matrix multiplication with a vector from the right is only defined, if the number of columns in A is equal to the dimensionality of the vector x.

Given a matrix of shape (n, d) and a vector of dimensionality n, the multiplication $x^\top A$ of them (with $x^\top$ being on the left hand side) is defined as a vector of length d such that

$$
\begin{aligned}
x^\top A &= \left[x_0, x_1, \ldots, x_{n-1}\right] \begin{bmatrix} a_{0,0} & \cdots & a_{0,d-1} \\ a_{1,0} & \cdots & a_{1,d-1} \\ \vdots & \ddots & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,d-1} \end{bmatrix} \\
&= \begin{bmatrix} x \cdot \begin{bmatrix} a_{0,0} \\ \vdots \\ a_{n-1,0} \end{bmatrix} & x \cdot \begin{bmatrix} a_{0,1} \\ \vdots \\ a_{n-1,1} \end{bmatrix} & \cdots & x \cdot \begin{bmatrix} a_{0,d-1} \\ \vdots \\ a_{n-1,d-1} \end{bmatrix} \end{bmatrix}
\end{aligned}
$$

Matrix-vector multiplication

Thus, $x^\top A$ is a vector and the k-th component of vector $x^\top A$ is given as an inner product

$$
\begin{aligned}
(x^\top A)_k &= x \cdot A[:, k] \\
&= \left(a_{0,k} \ldots a_{n-1,k}\right) \cdot x \\
&= \sum_{r=0}^{n-1} x_r a_{r,k}
\end{aligned}
$$

Matrix-vector multiplication

Consequence:

inner products as matrix vector multiplication

The inner product $x \cdot y$ of two column-shaped vectors can be written in matrix-vector multiplication notation as

$$x \cdot y = \begin{bmatrix} x_0, x_1, \ldots, x_{d-1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{d-1} \end{bmatrix} = x^\top y$$

where $x^\top$ is the transpose of a vector or matrix x.

The transpose was used here to convert the column-shaped vector into a row-shaped vector.

Matrix-Matrix multiplication

Given a matrix A of shape (n, d) and a matrix B of shape (d, f), their multiplication AB is defined as a matrix of shape (n, f) and its component $(AB)_{i,k}$ at row i and colunmn k is given as

$$(AB)_{i,k} = A_{(i,:)} \cdot B_{(:,k)} = \sum_{r=1}^{d} A_{i,r} B_{rk}$$

as an inner product between the i-th row of the left matrix and the $k-$th column of the right matrix.

Important: the number or dimensions in the second axis of A must be equal to the number or dimensions in the first axis of B. Otherwise matrix multiplication is not possible.

Matrix-Matrix multiplication

Therefore a matrix- matrix multiplication is a matrix consisting of inner products:

$$AB = \begin{bmatrix} A_{(0,:)} \cdot B_{(:,0)} & A_{(0,:)} \cdot B_{(:,1)} & A_{(0,:)} \cdot B_{(:,2)} & \cdots & A_{(0,:)} \cdot B_{(:,f-1)} \\ A_{(1,:)} \cdot B_{(:,0)} & A_{(1,:)} \cdot B_{(:,1)} & A_{(1,:)} \cdot B_{(:,2)} & \cdots & A_{(1,:)} \cdot B_{(:,f-1)} \\ A_{(2,:)} \cdot B_{(:,0)} & A_{(2,:)} \cdot B_{(:,1)} & A_{(2,:)} \cdot B_{(:,2)} & \cdots & A_{(2,:)} \cdot B_{(:,f-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{(n-1,:)} \cdot B_{(:,0)} & A_{(n-1,:)} \cdot B_{(:,1)} & A_{(n-1,:)} \cdot B_{(:,2)} & \cdots & A_{(n-1,:)} \cdot B_{(:,f-1)} \end{bmatrix}$$

Memorizing

- Matrix-Matrix multiplication results in a matrix, if shapes are permissible: $(n, d)(d, f) \to (n, f)$.
- The component $(AB)_{i,k}$ at row i and column k is given as the inner product between row i of the left matrix and column k of the right matrix.
  That also tells you which axes must match in dimensionality: left matrix - the number of columns $=$ the dimensionality of the second axis. right matrix - the number of rows $=$ the dimensionality of the first axis.

Matrix-Matrix multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

Matrix-Matrix multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ \color{blue}2 & \color{blue}4 & \color{blue}-1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} \color{blue}5 & -1 & -2 \\ \color{blue}3 & 9 & -3 \\ \color{blue}-2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ \color{blue}* & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ \color{blue}2 & \color{blue}4 & \color{blue}-1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & \color{blue}-1 & -2 \\ 3 & \color{blue}9 & -3 \\ -2 & \color{blue}1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & \color{blue}* & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ \color{blue}2 & \color{blue}4 & \color{blue}-1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & \color{blue}-2 \\ 3 & 9 & \color{blue}-3 \\ -2 & 1 & \color{blue}2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & \color{blue}* \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \\ -1 & 0 & 6 \end{bmatrix} \begin{bmatrix} 5 & -1 & -2 \\ 3 & 9 & -3 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

Learning goals

- Linear and affine equation systems (short)
- Solving Linear and affine equation systems using Gauss-Jordan algorithm
- Identifying the solutions of affine equation systems: no solution, a single solution, an affine space of solutions, based on the outcome of the Gauss-Jordan algorithm
- Extending a set of vectors to a basis of the whole space (by applying Gram-Schmid)

A note on pragmatical learning

like a python try: except: clause

- if you understand the explanation, it is best as it allows you to remember things easier, because you can deduct them again without memorizing too much – however this happens 10% all the time only
- if you do not understand the explanation, take a look at it after a sleep. The brain processes things from the previous days (cf. nightmares!)
- if you still do not understand the explanation, search for others
- if you still do not understand the explanation after trying n different of them, it is okay ... not everyone can be a Messi, a van Gogh, a supermodel, a gifted cook, same holds for math skills, then be pragmatical: memorize the rules how to apply it

Outline

Linear equation system

A set of n linear equations which are to be solved for the vector $x = (x_0, \ldots x_{d-1})$

$$a_0^{\{0\}} x_0 + a_1^{\{0\}} x_1 + \ldots + a_{d-1}^{\{0\}} x_{d-1} = a^{\{0\}} \cdot x = 0 \tag{1}$$
$$a_0^{\{1\}} x_0 + a_1^{\{1\}} x_1 + \ldots + a_{d-1}^{\{1\}} x_{d-1} = a^{\{1\}} \cdot x = 0 \tag{2}$$
$$\ldots \tag{3}$$
$$a_0^{\{n-1\}} x_0 + a_1^{\{n-1\}} x_1 + \ldots + a_{d-1}^{\{n-1\}} x_{d-1} = a^{\{n-1\}} \cdot x = 0 \tag{4}$$

is called a linear equation system

This is obviously a set of equations made from n inner products:

$$
\begin{aligned}
a^{\{0\}} \cdot x &= 0 \\
a^{\{1\}} \cdot x &= 0 \\
&\cdots \\
a^{\{n-1\}} \cdot x &= 0
\end{aligned}
$$

As an example, such inner products like here

$$(1,0,2,3) \cdot x = 0$$
$$(-1,1,0,1) \cdot x = 0$$

can be written in the sense of matrix-vector multiplication as:

$$\begin{bmatrix}1,0,2,3\end{bmatrix} x = 0$$
$$\begin{bmatrix}-1,1,0,1\end{bmatrix} x = 0$$

This in turn is the same as:

$$\begin{bmatrix}1, 0, 2, 3\end{bmatrix} \begin{bmatrix}x_0 \\ x_1 \\ x_2 \\ x_3\end{bmatrix} = 0$$

$$\begin{bmatrix}-1, 1, 0, 1\end{bmatrix} \begin{bmatrix}x_0 \\ x_1 \\ x_2 \\ x_3\end{bmatrix} = 0$$

Now one can fuse this into a single matrix vector multiplication:

$$\begin{bmatrix}1 & 0 & 2 & 3 \\ -1 & 1 & 0 & 1\end{bmatrix} \begin{bmatrix}x_0 \\ x_1 \\ x_2 \\ x_3\end{bmatrix} = 0$$

This conforms to the equation

$$Ax = A \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\text{with } A = \begin{bmatrix} 1 & 0 & 2 & 3 \\ -1 & 1 & 0 & 1 \end{bmatrix}$$

You can do this with more than two equations:

$$\begin{aligned}(1, 0, 2, 3) \cdot x &= 0\\(-1, 1, 0, 1) \cdot x &= 0\\(4, 3, 2, 1) \cdot x &= 0\end{aligned}$$

is equivalent to

$$Ax = A \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\text{with } A = \begin{bmatrix} 1 & 0 & 2 & 3 \\ -1 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Here with n equations:

$$\begin{aligned} w^{\{0\}} \cdot x &= 0 \\ w^{\{1\}} \cdot x &= 0 \\ \vdots &= 0 \\ w^{\{n-1\}} \cdot x &= 0 \end{aligned}$$

is equivalent to

$$Ax = 0$$

with $A = \begin{bmatrix} w^{\{0\}} \\ w^{\{1\}} \\ \vdots \\ w^{\{n-1\}} \end{bmatrix}$ where each $w^{\{i\}}$ is a horizontal / row vector.

Linear equation system II

A linear equation system can be defined by

$$
\begin{aligned}
Ax &= 0 \\
A.shape &= (n, d) \\
x.shape &= (d, 1)
\end{aligned}
$$

where one solves for the vector $x = (x_0, \ldots x_{d-1})$.

Affine equation system

A set of n affine equations which are to be solved for the vector $x = (x_0, \ldots x_{d-1})$

$$
\begin{aligned}
a_0^{\{0\}} x_0 + a_1^{\{0\}} x_1 + \ldots + a_{d-1}^{\{0\}} x_{d-1} &= a^{\{0\}} \cdot x = b_0 \\
a_0^{\{1\}} x_0 + a_1^{\{1\}} x_1 + \ldots + a_{d-1}^{\{1\}} x_{d-1} &= a^{\{1\}} \cdot x = b_1 \\
&\ldots \\
a_0^{\{n-1\}} x_0 + a_1^{\{n-1\}} x_1 + \ldots + a_{d-1}^{\{n-1\}} x_{d-1} &= a^{\{n-1\}} \cdot x = b_{n-1}
\end{aligned}
$$

is called an affine equation system. b_i are called the bias terms.

This is equivalent to

$$Ax = b$$

$$\text{with } A = \begin{bmatrix} w^{\{0\}} \\ w^{\{1\}} \\ \vdots \\ w^{\{n-1\}} \end{bmatrix}, \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

where each $w^{\{i\}}$ is a horizontal / row vector, that is of shape $(1, d)$.

Outline

The first step is to find a set of operations which do not change the solution. That is, we want to perform an operation L by matrix multiplication such that

$$Ax = b$$
$$LAx = Lb$$

have the same set of solutions x.

Swapping two rows

- This just swaps equations. So it does not affect the solutions.

Multiplying a row with a non-zero value

- if $a \neq 0$, then

$$
\begin{aligned}
v_1 x_1 + v_2 x_2 + \ldots + v_{n-1} x_{n-1} &= v \cdot x &= b \\
\text{and } a v_1 x_1 + a v_2 x_2 + \ldots + a v_{n-1} x_{n-1} &= a v \cdot x &= ab
\end{aligned}
$$

have the same solutions x.

Which operations can we use to solve? Adding a multiple of a row to another row

We consider the following: Equation set S1 is defined as:

$$a^{\{i\}} \cdot x = b_i$$
$$a^{\{k\}} \cdot x = b_k$$

Equation set S2 is defined as:

$$a^{\{i\}} \cdot x = b_i$$
$$(a^{\{k\}} + \beta a^{\{i\}}) \cdot x = b_k + \beta b_i$$

The S2 was created from S1 by adding a multiple of the first equation (multiplied with β) to the second equation.

Adding a multiple of an equation to another does not change the solutions

If we take any two equations from an affine equation system, and we add a multiple of an equation, then we do not change the set of solution vectors x. The solution for the original set, and for the set with added equations are the same.

Why is this true?

Adding a multiple of an equation to another does not change the solutions

If we take any two equations from an affine equation system, and we add a multiple of an equation, then we do not change the set of solution vectors x. The solution for the original set, and for the set with added equations are the same.

Suppose z solves S1, which means:

$$a^{\{i\}} \cdot z = b_i$$
$$a^{\{k\}} \cdot z = b_k$$

But then also this holds:

$$\beta a^{\{i\}} \cdot z = \beta b_i$$

... Now just add these equations

Suppose z solves S1, which means:

$$a^{\{i\}} \cdot z = b_i$$
$$a^{\{k\}} \cdot z = b_k$$

But then also this holds:

$$\beta a^{\{i\}} \cdot z = \beta b_i$$

Adding the first equation from S1 to the last equation results in S2

$$a^{\{i\}} \cdot z = b_i$$
$$(a^{\{k\}} + \beta a^{\{i\}}) \cdot z = b_k + \beta b_i$$

So we have shown, if z solves S1, then z also solves S2.

The other way round can be also shown easily: If z solves S2, then these two equations hold:

$$a^{\{i\}} \cdot z = b_i$$
$$(a^{\{k\}} + \beta a^{\{i\}}) \cdot z = b_k + \beta b_i$$

Now add $-\beta$ times the first equation (which is satisfied by assumption that z solves the equation set S2) to the second equation. This transforms S2 into

$$a^{\{i\}} \cdot z = b_i$$
$$(a^{\{k\}} + \beta a^{\{i\}}) \cdot z - \beta a^{\{i\}} \cdot z = b_k + \beta b_i - \beta b_i$$

which is S1:

$$a^{\{i\}} \cdot z = b_i$$
$$a^{\{k\}} \cdot z = b_k$$

Which operations can we use to solve? Adding a multiple of a row to another row

Conclusion:

- We have shown that adding a multiple of a row to another row does not change the solutions of an affine equation $Ax = b$

This method is suitable for solving small systems by hand

- Step 1: rewrite the affine equation system

$$Ax = b$$

into to the augmented matrix

$$\left[A|b\right] = \left[\begin{array}{ccccc|c} a_{00} & a_{01} & a_{02} & \cdots & a_{0,d-1} & b_0 \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,d-1} & b_1 \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2,d-1} & b_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,0} & a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,d-1} & b_{n-1} \end{array}\right]$$

- Step 2: obtain the so-called row echelon form.
  To understand the idea behind a row echelon form, let us consider at first a special case:
  If we had a (d, d)-shaped matrix A, then , in the second step, we want to achieve at first an upper diagonal matrix. It could look like this one:

$$A = \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ 0 & 0 & 1 & \dots & * & * \\ 0 & 0 & 0 & \dots & * & * \\ \vdots & \vdots & \vdots & \ddots & 1 & * \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

  then, in Step 3, we process it into a diagonal matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Why ? If the augmented matrix looks like that,

$$A = \left[\begin{array}{cccccc|c} 1 & 0 & 0 & \dots & 0 & 0 & b_0 \\ 0 & 1 & 0 & \dots & 0 & 0 & b_1 \\ 0 & 0 & 1 & \dots & 0 & 0 & b_2 \\ 0 & 0 & 0 & \dots & 0 & 0 & b_3 \\ \vdots & \vdots & \vdots & \ddots & 1 & 0 & b_* \\ 0 & 0 & 0 & \dots & 0 & 1 & b_{n-1} \end{array}\right]$$

then the resulting solution is

$$\begin{aligned} x_0 &= b_0 \\ x_1 &= b_1 \\ &\vdots \\ x_{n-1} &= b_{n-1} \end{aligned}$$

that is, we can read off the solution right away.

The shape of matrix which we want to achieve in the step 2, if the matrix A is any matrix, is a so-called **row echelon form:**

Definition: Row echelon form

Definition: Row echelon form We look only at the part without the bias column $b_0, b_1, b_2, \ldots$.

- All rows consisting of only zeroes are at the bottom of the matrix.
- for any non-zero row: The left-most nonzero entry of this row ('leading entry of a row') is to the right of the left-most nonzero entry of every row above it.
- the leading entry in any non-zero row is 1

Examples:

$$A = \begin{bmatrix} 1 & f & c \\ 0 & 1 & g \\ 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 1 & f & c \\ 0 & 1 & e \\ 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & c \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & a & b & c \\ 0 & 1 & d & e \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 1 & a & c & b \\ 0 & 1 & f & d \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & c & f \\ 0 & 0 & 1 & e \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & b & c & d \\ 0 & 0 & 1 & g \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & e & d \\ 0 & 1 & c & g \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & e & d \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Row Echelon Form Criteria:

We look only at the part without the bias column $b_0, b_1, b_2, \ldots$.

- All rows consisting of only zeroes are at the bottom of the matrix.
- for any non-zero row: The left-most nonzero entry of this row ('leading entry of a row') is to the right of the left-most nonzero entry of every row above it.
- the leading entry in any non-zero row is 1

What would be **NOT** a row echelon form by violating the second statement ? Here a counterexample:

$$Z = \begin{bmatrix} 1 & x & e & d \\ 0 & 1 & c & g \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} x & y & e & d \\ 1 & z & c & g \\ 0 & 1 & f & g \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The violation occurs in which row?

Row Echelon forms are a special case of upper diagonal matrices:

Definition upper diagonal matrix

B is an upper diagonal matrix, if all entries below the main diagonal are zero. That is, if

$$i > k \Rightarrow B_{ik} = 0$$

The entries below the main diagonal have a larger row index than their column index (B_{ik}, $i > k$).

- You can use the following operations to turn $\left[A|b\right]$ into the row echelon form:
  - swapping two rows (including the b_i-term)
  - multiplying a whole row (including the b_i-term) with a non-zero number
  - Adding a multiple of one row to another row (including the b_i-term)
- We know that all these operations do not change the solution space.

- Step 3: further process the row echelon form to a reduced row echelon form:
  Goal is a matrix in which every column, which contains a left-most non-zero entry, has only this entry as non-zero entry. This is a so-called **Reduced row echelon form**.

> **Definition: Reduced Row echelon form**
>
> Definition: Reduced Row echelon form
>
> - It is a row echelon form
> - all other entries in a column containing a left-most non-zero entry are 0

- note: Columns which have no left-most non-zero entry cannot be further processed.

Examples for the the reduced row echelon form:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & e \\ 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & c & 0 \\ 0 & 1 & f & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & b & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & b & 0 & d \\ 0 & 0 & 1 & g \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & e & d \\ 0 & 1 & c & g \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & e & 0 \\ 0 & 1 & c & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

We obtain the reduced row echelon form from a matrix in row echelon form by adding rows with left-most non-zero entries to columns above them:

$$A = \begin{bmatrix} 1 & a & b & c & g \\ 0 & 1 & d & e & h \\ 0 & 0 & 1 & f & i \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \to \widetilde{A} = \begin{bmatrix} 1 & 0 & 0 & \widetilde{c} & 0 \\ 0 & 1 & 0 & \widetilde{e} & 0 \\ 0 & 0 & 1 & \widetilde{f} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
\left[\begin{array}{ccc|c} 3 & 3 & 3 & 6 \\ 2 & -1 & -4 & 13 \\ -3 & 2 & 2 & -20 \end{array}\right] \to (r_0 *= 1/3) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 2 & -1 & -4 & 13 \\ -3 & 2 & 2 & -20 \end{array}\right]
$$

$$
\to (r_1 += -2r_0), (r_2 += 3r_0) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & -3 & -6 & 9 \\ 0 & 5 & 5 & -14 \end{array}\right]
$$

$$
\to (r_1 *= 1/-3) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & -3 \\ 0 & 5 & 5 & -14 \end{array}\right]
$$

$$
\to (r_2 += -5r_1) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & -3 \\ 0 & 0 & -5 & 1 \end{array}\right]
$$

$$
\to (r_2 *= -5) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & -3 \\ 0 & 0 & 1 & -1/5 \end{array}\right]
$$

Example with a row swap:

$$\left[\begin{array}{ccc|c} 2 & 4 & 6 & 5 \\ 3 & 6 & -3 & 4.5 \\ -2 & 2 & 2 & 7 \end{array}\right] \to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2.5 \\ 3 & 6 & -3 & 4.5 \\ -2 & 2 & 2 & 7 \end{array}\right]$$

$$A_{1,:} = A_{1,:} - 3A_{0,:} \to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2.5 \\ 0 & 0 & -12 & -3 \\ 0 & 6 & 8 & 12 \end{array}\right]$$

$$\text{swap here} \to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2.5 \\ 0 & 6 & 8 & 12 \\ 0 & 0 & -12 & -3 \end{array}\right]$$

$$\to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2.5 \\ 0 & 1 & 4/3 & 2 \\ 0 & 0 & -12 & -3 \end{array}\right]$$

$$\to \left[\begin{array}{ccc|c} 1 & 2 & 3 & 2.5 \\ 0 & 1 & 4/3 & 2 \\ 0 & 0 & 1 & 1/4 \end{array}\right]$$

- swap all zero rows to the bottom
- For the 0-th row:
  - If $a_{0,0}$ is zero, follow this general idea:

general idea of finding the leftmost non-zero entry

find in the current row of interest “the leftmost non-zero entry”, such that there is ...
no non-zero entry which is at the same time further lower <u>and</u> further left of it.
You can swap rows to find this state.

Examples:

$$A = \left[\begin{array}{cccc|c} 0 & \underline{c} & d & e & b_0 \\ 0 & g & h & f & b_1 \\ \underline{r} & l & m & k & b_2 \\ \underline{s} & n & o & p & b_3 \end{array}\right] \text{ focus on zero-th row: not ok case !}$$

$$\rightarrow A = \left[\begin{array}{cccc|c} \underline{r} & l & m & k & b_2 \\ 0 & c & d & e & b_0 \\ 0 & g & h & f & b_1 \\ s & n & o & p & b_3 \end{array}\right] \text{ focus on zero-th row: okay case !}$$

This was obtained by swapping the zero-th and the second row!

Examples:

$$A = \left[\begin{array}{cccc|c} 0 & 0 & \underline{d} & e & b_0 \\ 0 & \underline{c} & h & f & b_1 \\ 0 & \underline{l} & m & k & b_2 \\ 0 & \underline{n} & o & p & b_3 \end{array}\right]$$ focus on zero-th row: not ok case !

$$A = \left[\begin{array}{cccc|c} 0 & \underline{c} & h & f & b_1 \\ 0 & 0 & d & e & b_0 \\ 0 & l & m & k & b_2 \\ 0 & n & o & p & b_3 \end{array}\right]$$ focus on zero-th row: okay case !

$$A = \left[\begin{array}{cccc|c} 0 & 0 & \underline{d} & e & b_0 \\ 0 & 0 & h & f & b_1 \\ 0 & \underline{l} & m & k & b_2 \\ 0 & \underline{n} & o & p & b_3 \end{array}\right]$$ focus on zero-th row: not ok case !

$$A = \left[\begin{array}{cccc|c} 0 & 0 & \underline{d} & e & b_0 \\ 0 & 0 & h & f & b_1 \\ 0 & 0 & m & k & b_2 \\ 0 & 0 & o & p & b_3 \end{array}\right]$$ focus on zero-th row: okay case !

Can swap rows if needed:

$$A = \left[\begin{array}{cccc|c} 0 & c & d & e & b_0 \\ 0 & g & h & f & b_1 \\ r & l & m & k & b_2 \\ s & n & o & p & b_3 \end{array}\right] \rightarrow A = \left[\begin{array}{cccc|c} r & l & m & k & b_2 \\ 0 & c & d & e & b_0 \\ 0 & g & h & f & b_1 \\ s & n & o & p & b_3 \end{array}\right]$$

- If $a_{0,0}$ is zero, swap rows from the bottom of it, until $a_{0,0} \neq 0$.
- If getting $a_{0,0} \neq 0$ by swapping rows is not possible, it means that the whole 0-th column $A_{0:end,0}$ must be zero. In that case do the same thing with one column to the right: check if $a_{0,1}$ is zero, swap rows from the bottom of it, until $a_{0,1} \neq 0$.

$$A = \left[\begin{array}{cccc|c} 0 & 0 & d & e & b_0 \\ 0 & g & h & f & b_1 \\ 0 & l & m & k & b_2 \\ 0 & n & o & p & b_3 \end{array}\right] \rightarrow A = \left[\begin{array}{cccc|c} 0 & g & h & f & b_1 \\ 0 & 0 & d & e & b_0 \\ 0 & l & m & k & b_2 \\ 0 & n & o & p & b_3 \end{array}\right]$$

- If that is not possible, then the first column is all zero, too. Then do the same with one column to the right further : check if $a_{0,2}$ is zero ...

$$A = \begin{bmatrix} 0 & 0 & 0 & e & | & b_0 \\ 0 & 0 & h & f & | & b_1 \\ 0 & 0 & m & k & | & b_2 \\ 0 & 0 & z & p & | & b_3 \end{bmatrix} \rightarrow A = \begin{bmatrix} 0 & 0 & h & f & | & b_1 \\ 0 & 0 & 0 & e & | & b_0 \\ 0 & 0 & m & k & | & b_2 \\ 0 & 0 & z & p & | & b_3 \end{bmatrix}$$

Suppose you have found a leftmost $a_{0,r_0} \neq 0$.

- Suppose you have found a leftmost $a_{0,r_0} \neq 0$. r_0 can be at index 0 if we have $a_{0,0} \neq 0$.
- next step: multiply the 0-th row so that $a_{0,r_0} = 1$. do the same operation with the same multiplier to the bias column.

$$A = \begin{bmatrix} a & b & c & d & | & b_0 \\ e & f & g & h & | & b_1 \\ i & k & l & m & | & b_2 \\ n & o & q & q & | & b_3 \end{bmatrix} \to \widehat{A} = \begin{bmatrix} \underline{1} & \widehat{b} & \widehat{c} & \widehat{d} & | & \widehat{b_0} \\ e & f & g & h & | & b_1 \\ i & k & l & m & | & b_2 \\ n & o & q & q & | & b_3 \end{bmatrix}$$

Part 2: Getting to the row echelon form in general

- Add multiples of the 0-th row $a_{0,:}$ to all rows, such that the r_0-column in all other rows becomes zero. Do the same operation with the same multiplier to the bias column.

$$A = \left[\begin{array}{cccc|c} 1 & b & c & d & b_0 \\ e & f & g & h & b_1 \\ i & k & l & m & b_2 \\ n & o & p & q & b_3 \end{array}\right] \to \widehat{A} = \left[\begin{array}{cccc|c} 1 & b & c & d & b_0 \\ 0 & \widehat{f} & \widehat{g} & \widehat{h} & \widehat{b_1} \\ 0 & \widehat{k} & \widehat{l} & \widehat{m} & \widehat{b_2} \\ 0 & \widehat{o} & \widehat{p} & \widehat{q} & \widehat{b_3} \end{array}\right] \begin{array}{c} \downarrow \cdots\cdots\cdots \downarrow \\ \text{add (row 0)*factor(-e) to row 1} \\ \text{add (row 0)*factor(-i) to row 2} \\ \text{add (row 0)*factor(-n) to row 3} \end{array}$$

or, if the 0-th column was all zeros and your $r_0 = 1$:

$$A = \left[\begin{array}{cccc|c} 0 & 1 & c & d & b_0 \\ 0 & f & g & h & b_1 \\ 0 & k & l & m & b_2 \\ 0 & o & p & q & b_3 \end{array}\right] \to \widehat{A} = \left[\begin{array}{cccc|c} 0 & 1 & c & d & b_0 \\ 0 & 0 & \widehat{g} & \widehat{h} & \widehat{b_1} \\ 0 & 0 & \widehat{l} & \widehat{m} & \widehat{b_2} \\ 0 & 0 & \widehat{p} & \widehat{q} & \widehat{b_3} \end{array}\right]$$

Now repeat the same three steps for the row number 1:

- Find $A_{1,r_1} \neq 0$ such that there is no non-zero entry in rows below which is further left of column index r_1. You can swap rows to achieve this
- Multiply the row with $\frac{1}{A_{1,r_1}}$
- Add multiples of the row to all rows below so that all entries in the column indexed with r_0 become zero: $A_{2,r_1} = 0, \ldots, A_{n-1,r_1} = 0$

$$A = \begin{bmatrix} 1 & b & c & d & | & b_0 \\ 0 & 1 & g & h & | & b_1 \\ 0 & k & l & m & | & b_2 \\ 0 & o & p & q & | & b_3 \end{bmatrix} \rightarrow \widehat{A} = \begin{bmatrix} 1 & b & c & d & | & b_0 \\ 0 & 1 & g & h & | & b_1 \\ 0 & 0 & \widehat{l} & \widehat{m} & | & \widehat{b_2} \\ 0 & 0 & \widehat{p} & \widehat{q} & | & \widehat{b_3} \end{bmatrix} \begin{array}{l} \downarrow \cdots\cdots\cdots \downarrow \\ \text{add (row 1)*factor(-k) to row 2} \\ \text{add (row 1)*factor(-o) to row 3} \end{array}$$

Now repeat the same three steps for all other row numbers k:

- Find $A_{k,r_k} \neq 0$ such that there is no non-zero entry in rows below which is further left of column index r_k. You can swap rows to achieve this
- Multiply the row with $\frac{1}{A_{k,r_k}}$
- Add multiples of the row to all rows below so that all entries in the column indexed with r_0 become zero: $A_{k+1,r_k} = 0, \dots, A_{n-1,r_k} = 0$

$$A = \left[\begin{array}{cccccc|c} 1 & 0 & * & * & \dots & * & * \\ 0 & 1 & * & * & \dots & * & * \\ 0 & 0 & 1 & * & \dots & * & * \\ 0 & 0 & * & * & \dots & * & * \\ 0 & 0 & * & * & \dots & * & * \\ 0 & 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & * & * & \dots & * & * \end{array}\right] \rightarrow A = \left[\begin{array}{cccccc|c} 1 & * & * & * & \dots & * & * \\ 0 & 1 & * & * & \dots & * & * \\ 0 & 0 & 1 & * & \dots & * & * \\ 0 & 0 & 0 & * & \dots & * & * \\ 0 & 0 & 0 & * & \dots & * & * \\ 0 & 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & * & \dots & * & * \end{array}\right]$$

- this is much simpler: you go through all columns which have a left-most-non zero entry and make all other entries in this column turn to zero.
- you use only adding rows to others. Difference to before: you add a row to rows above of it.
- you do less operations if you go through all columns which have a left-most-non zero entry – starting from the right. But it also works, if you would start from the left as you have done it in **main step 2**

$$\begin{bmatrix} 1 & 1 & 1 & | & 2 \\ 0 & 1 & 2 & | & -3 \\ 0 & 0 & 1 & | & -1/5 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & | & 2 \\ 0 & 1 & 0 & | & -3+2/5=-2.4 \\ 0 & 0 & 1 & | & -1/5 \end{bmatrix} \begin{array}{c} \\ \text{add row 2 to row 1} \\ \uparrow \cdots \uparrow \end{array}$$

$$\to \begin{bmatrix} 1 & 1 & 0 & | & 2+1/5=2.2 \\ 0 & 1 & 0 & | & -2.4 \\ 0 & 0 & 1 & | & -1/5 \end{bmatrix} \begin{array}{c} \text{add row 2 to row 0} \\ \\ \uparrow \cdots \uparrow \end{array}$$

- this is much simpler: you go through all columns which have a left-most-non zero entry and make all other entries in this column turn to zero.
- you use only adding rows to others. Difference to before: you add a row to rows above of it.
- you do less operations if you go through all columns which have a left-most-non zero entry – starting from the right. But it also works, if you would start from the left as you have done it in **main step 2**

$$\begin{bmatrix} 1 & 1 & 0 & | & 2+1/5=2.2 \\ 0 & 1 & 0 & | & -2.4 \\ 0 & 0 & 1 & | & -1/5 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & | & 2.2 - -2.4 = 4.6 \\ 0 & 1 & 0 & | & -2.4 \\ 0 & 0 & 1 & | & -1/5 \end{bmatrix} \begin{matrix} \text{add row 1 to row 0} \\ \uparrow \cdots \uparrow \\ \ \end{matrix}$$

You are adding whole rows. I only put focus on what is important by row of origin change to zero change to any value. If there would be a fourth column, as in the following example, then it would change too:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & | & 2 \\ 0 & 1 & 2 & 3 & | & -3 \\ 0 & 0 & 1 & -2 & | & -1/5 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 1 & | & 2 \\ 0 & 1 & 0 & 7 & | & -3+2/5=-2.4 \\ 0 & 0 & 1 & -2 & | & -1/5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & | & 2 \\ 0 & 1 & 0 & 7 & | & -2.4 \\ 0 & 0 & 1 & -2 & | & -1/5 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 0 & 3 & | & 2+1/5=2.2 \\ 0 & 1 & 0 & 7 & | & -2.4 \\ 0 & 0 & 1 & -2 & | & -1/5 \end{bmatrix}$$

Observe: Anything in columns left of the left-most-non zero entry will not be changed

When A is of shape $d \times d$ and has full rank, then you will obtain an equation which looks like this:

$$A = \left[\begin{array}{ccccc|c} 1 & 0 & 0 & \dots & 0 & b_0 \\ 0 & 1 & 0 & \dots & 0 & b_1 \\ 0 & 0 & 1 & \dots & 0 & b_2 \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & b_{d-1} \end{array}\right]$$

This corresponds to a directly read-off solution

$$\begin{aligned} x_0 &= b_0 \\ x_1 &= b_1 \\ x_2 &= b_2 \\ &\vdots \\ x_{d-1} &= b_{d-1} \end{aligned}$$

If you obtain a row, where all entries are zeros, but the bias is not zero, $b \neq 0$,

$$A = \left[\begin{array}{cccccc|c} 1 & 0 & * & * & \dots & * & * \\ 0 & 1 & * & * & \dots & * & * \\ 0 & 0 & 0 & 0 & \dots & 0 & b \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right]$$

then you know that this has no solution: This corresponds to trying to solve

$$0x_0 + 0x_1 + \dots + 0x_{d-1} = b \neq 0$$

This is common to be observed when $n > d$. That is when you have more equations n than dimensions d in your solution space.

$\boxed{\text{In this case: a row with all zeros in the matrix part, and a non-zero bias } b \neq 0\text{, no solution exists.}}$

What solutions can we get? A case which does not affect solutions

If you obtain such a condition with a zero bias $b = 0$

$$A = \left[\begin{array}{cccccc|c} 1 & 0 & * & * & \ldots & * & b_0 \\ 0 & 1 & * & * & \ldots & * & b_1 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{array}\right],$$

it is a row stating

$$0x_0 + 0x_1 + \ldots + 0x_{d-1} = 0$$

It poses no constraint on the solution. This is satisfied by any values for $x_0, x_1, \ldots, x_{d-1}$. Therefore you can remove or ignore such a row.

Zero rows in A can also happen when $n \leq d$, and some of the rows of A are a linear combination of other rows of A. Then Gauss-Jordan transforms these rows into zero-rows.
Example for a case with $n \leq d$ and zero rows in A after transformation:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

What solutions can we get? Rows with zero bias and conditions with many solutions

If you obtain, after removing conditions with zero bias, a structure like that:

$$A = \left[\begin{array}{cccccc|c} 1 & 0 & * & * & \dots & * & b_0 \\ 0 & 1 & * & * & \dots & * & b_1 \end{array}\right],$$

then you have as solution an affine space with as many dimensions, as columns with stars $*$.
How to solve it ?

- If you just need any solution, then the simplest way is to plug in a zero $(= 0)$ for all the variables with a $*$.

In the above example:

$$\begin{aligned} x_0 &= b_0 \\ x_1 &= b_1 \\ x_2 &= x_3 = \dots = x_{d-1} = 0 \end{aligned}$$

What solutions can we get? Rows with zero bias and conditions with many solutions

- For other solutions: You can plug in any values for the variables with a $*$, and move the resulting numbers to the existing bias term. Then solve for the terms having a 1 coefficient.

Example here:

$$A = \begin{bmatrix} 1 & 0 & -2 & 3.5 & | & 5 \\ 0 & 1 & 3 & -1 & | & 7 \end{bmatrix}$$

You could plug in $x_2 = 4$. This results in $-2 * 4$ and $3 * 4$ for the column corresponding to x_2. You could plug in $x_3 = 2$. This results in $3.5 * 2$ and $-1 * 2$ for the column corresponding to x_3. Bring this to the side of the bias:

$$x_2 = 4, \; x_3 = 2 \text{ and}$$
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & | & 5 - (-2 * 4) - 3.5 * 2 \\ 0 & 1 & 0 & 0 & | & 7 - (3 * 4) - (-1 * 2) \end{bmatrix} \Rightarrow x_0 = 6, x_1 = -3$$

Identifying solutions from the reduced row echelon form

The solution to the affine equation

One can see this by looking at the reduced row echelon form after processing $Ax = b$

- it can have no solution, a single solution, or an affine space of solutions.
- it has no solution, if the Gauss-Jordan Algorithm results (at any step) in a matrix where one row has only zeros in its matrix part, but the corresponding bias term of that row is non-zero
- special case: it has exactly one solution if A is of shape (d, d) and the Gauss-Jordan Algorithm in Main step 2 results in an upper diagonal matrix with only non-zero entries on the diagonal
- it has a vector space of solutions if the result of the Gauss-Jordan Algorithm shows that
  - all rows which have only zeros in their matrix part have also their bias term being zero.
  - there exists a column which has no left-most non-zero entry
- it has exactly one solution if the result of the Gauss-Jordan Algorithm shows that
  - all rows which have only zeros in their matrix part have also their bias term being zero.
  - all columns have a left-most non-zero entry

These cases have also certain namings for the equation systems

- no solution: inconsistent equation system
- a single solution: consistent and determinate equation system
- an affine space of solutions:consistent and indeterminate equation system

An affine space is a vector space with an added offset vector, that is:

$$A = \{a + v, v \in V\}$$

where V is a vector space.
An affine space can be described in terms of a basis $v_0, \ldots, v_{d-1}$ of a vector space V the set of

$$A = \{a + \sum_{k=0}^{d-1} b_k v_k, b_k \in \mathbb{R}\}$$

Why the solutions of this define an affine space?

$$A = \begin{bmatrix} 1 & 0 & -2 & 3.5 & | & 5 \\ 0 & 1 & 3 & -1 & | & 7 \end{bmatrix},$$

Lets write them out:

$$\begin{aligned} x_0 - 2x_2 + 3.5x_3 &= 5 \\ x_1 + 3x_2 - 1x_3 &= 7 \end{aligned}$$

This transforms into:

$$\begin{aligned} x_0 &= 5 + 2x_2 - 3.5x_3 \\ x_1 &= 7 - 3x_2 + 1x_3 \end{aligned}$$

Now write this in terms of a vector

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Affine space

$$x_0 = 5 + 2x_2 - 3.5x_3$$
$$x_1 = 7 - 3x_2 + 1x_3$$
$$\Rightarrow \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ -3 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -3.5 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

This conforms to the definition of an affine space

$$A = \{x = a + b_0 v_0 + b_1 v_1, b_0 \in \mathbb{R}, b_1 \in \mathbb{R}\} \text{ with}$$
$$a = \begin{bmatrix} 5 \\ 7 \\ 0 \\ 0 \end{bmatrix}, b_0 = x_2, v_0 = \begin{bmatrix} 2 \\ -3 \\ 1 \\ 0 \end{bmatrix}, b_1 = x_3, v_1 = \begin{bmatrix} -3.5 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

The next sections are about understanding what the solutions are for linear and affine equation systems. At first we need a helping construction when we are given a set of vectors $w^{\{0\}}, \ldots, w^{\{k-1\}}$.

$\boxed{\text{What is our goal here ?}}$

- We start with a given set of k vectors $w^{\{0\}}, \ldots, w^{\{k-1\}}$. The space has dimensionality d.
- We want to add $d-k$ vectors $w^{\{k\}}, w^{\{k+1\}}, \ldots, w^{\{d-1\}}$ such that

(Prop 1) the union of all vectors $w^{\{0\}}, \ldots, w^{\{k-1\}}, w^{\{k\}}, w^{\{k+1\}}, \ldots, w^{\{d-1\}}$ are a basis of $\mathbb{R}^d$
Rmb: Basis means independent + every vector from $\mathbb{R}^d$ can be expressed as a linear combination of the basis vectors

(Prop 2) any vector from $w^{\{0\}}, \ldots, w^{\{k-1\}}$ (the first part) is orthogonal to any vector from $w^{\{k\}}, w^{\{k+1\}}, \ldots, w^{\{d-1\}}$ (the second part). Mathematically one can write this condition as:

$$r \leq k-1, s \geq k \Rightarrow w^{\{r\}} \cdot w^{\{s\}} = 0$$

$$r \leq k-1, s \geq k \Rightarrow w^{\{r\}} \cdot w^{\{s\}} = 0$$

insight

Any vector, which is a linear combination from the first part, is orthogonal to any vector, which is a linear combination from the second part.

One can also say that the vectors from the first part create a vector space which is orthogonal to the vector space created by the vectors from the second part.

Out of exams: Why is this true?

$$u = \sum_{r=1}^{k-1} a_r w^{\{r\}} \text{ see: summing of } r \text{ runs only over the first part}$$

$$v = \sum_{s=k}^{d-1} a_s w^{\{s\}} \text{ see: summing of } r \text{ runs only over the 2nd part}$$

$$\begin{aligned}
u \cdot v &= (\sum_{r=1}^{k-1} a_r w^{\{r\}}) \cdot (\sum_{s=k}^{d-1} a_s w^{\{s\}}) \\
&= \sum_{r=1}^{k-1} a_r (w^{\{r\}} \cdot (\sum_{s=k}^{d-1} a_s w^{\{s\}})) \text{ use linearity of inner product in its components!} \\
&= \sum_{r=1}^{k-1} a_r \sum_{s=k}^{d-1} a_s \underbrace{(w^{\{r\}} \cdot w^{\{s\}})}_{=0} = 0 \text{ due to } r \leq k-1, s \geq k
\end{aligned}$$

Examples for such extension of a set of vectors:

- We start with the vector $(1, 0, 0)$ (spans the space of all $(a, 0, 0)$). We can extend $(1, 0, 0)$ by $(0, 1, 0)$ and $(0, 0, 1)$. These three vectors allow to represent any vector $(a_0, a_1, a_2) \in \mathbb{R}^3$
- We can extend $(1, 0, 0)$ also by $(0, 1, -1)$ and $(0, 1, 1)$
- We can extend $(1, 1, 0)$ by $(-2, 2, 0)$ and $(0, 0, 1)$

How to get a set with Prop1 and Prop2?

Gram-Schmid will be the tool to achieve that extension of an independent set!

Remember that Gram-Schmid results in an orthogonal set of vectors $\widetilde{v}^{\{0\}}, \ldots, \widetilde{v}^{\{d-1\}}$, if the input is an independent set, that is $\widetilde{v}^{\{r\}} \cdot \widetilde{v}^{\{s\}} = 0$ whenever $r \neq s$

- we know that the d-dimensional space of real valued vectors has this set of vectors as basis:

$$e^{\{0\}} = \underbrace{(1, 0, \dots, 0)}_{\text{d of them}}, e^{\{1\}} = (0, 1, 0, \dots, 0), e^{\{2\}} = (0, 0, 1, \dots, 0), e^{\{d-1\}} = (0, 0, 0, \dots, 1)$$

These are one hot-vectors. They are zero, except at a single position, where they are one.

- start with this initialization:

$$v^{\{0\}} = w^{\{0\}}, v^{\{1\}} = w^{\{1\}}, \dots, v^{\{k-1\}} = w^{\{k-1\}}, \text{ k vecs}$$
$$v^{\{k\}} = e^{\{0\}}, v^{\{k+1\}} = e^{\{1\}}, \dots, v^{\{k+d-1\}} = e^{\{d-1\}} \text{ d vecs}$$

that is we create a sequence which starts with $w^{\{0\}}, \dots, w^{\{k-1\}}$, then we add the other basis vectors to the sequence behind these k vectors.

- Next: we run Gram-Schmidt-Orthogonalization on this set.

Next: we run Gram-Schmidt-Orthogonalization on this set.

- Important 1: the vector space spanned by $v^{\{0\}}, \ldots, v^{\{k-1\}}$ will be the same as the space spanned by the outputs $\tilde{v}^{\{0\}}, \ldots, \tilde{v}^{\{k-1\}}$. See the last lecture for a proof.[1]
- Important 2: when we run Gram-Schmidt past index $k-1$, then it will remove the components of all vectors $v^{\{0\}}, \ldots, v^{\{k-1\}}$ (up to index $k-1$) from all the following vectors starting at index k, that is from $v^{\{k\}}, \ldots, v^{\{k+d-1\}}$.
- So the resulting $\tilde{v}^{\{k\}}, \ldots, \tilde{v}^{\{k+d-1\}}$ will be orthogonal to all vectors $v^{\{0\}}, \ldots, v^{\{k-1\}}$ – as we wanted to find!

[1] I did not prove that Gram-Schmidt does not increase the space of representable vectors.

An example an algorithm to obtain a basis with (Prop 1) and (Prop 2)

- We know that the space is d-dimensional - that is every basis set has exactly d vectors, but we have given $k+d$ vectors as input for the algorithm.
- From the first k vectors, none of the will become zero (because we assumed that they are linearly independent). Therefore k of the last d basis vectors $e^{\{0\}}, \ldots, e^{\{d-1\}}$ must turn into zero vectors after orthogonalization.

Solution: We will use $\widetilde{v}^{\{0\}}, \ldots, \widetilde{v}^{\{k-1\}}$ for the first part. We will use those $d-k$ vectors from the set $\widetilde{v}^{\{k\}}, \ldots, \widetilde{v}^{\{k+d-1\}}$ for the second part, which did not become zero vectors .

Together they are d orthogonal vectors, and therefore they must be a basis of the vector space $\mathbb{R}^d$ as desired.

Next step: We do the same, but for the case, when the input set of k vectors

$$w^{\{0\}}, \dots, w^{\{k-1\}}$$

is possibly not independent.

We do this very similarly to the previous application of Gram-Schmidt.

- start with this initialization:

$$v^{\{0\}} = w^{\{0\}}, v^{\{1\}} = w^{\{1\}}, \ldots, v^{\{k-1\}} = w^{\{k-1\}}, \text{ that are k vecs}$$
$$v^{\{k\}} = e^{\{0\}}, v^{\{k+1\}} = e^{\{1\}}, \ldots, v^{\{k+d-1\}} = e^{\{d-1\}}, \text{ that are d vecs}$$

  that is we create a sequence which starts with $w^{\{0\}}, \ldots, w^{\{k-1\}}$, then we add the other basis vectors to the sequence behind these k vectors.
- Next: we run Gram-Schmidt on this set.
- There is one difference:
  Since the first k vectors are not independent, there can be zeros among the output of the first k vectors. If this happens, then the last d vectors would have a few zero vectors less. So we have to look at the whole output (of length $k + d$) $\widetilde{v}^{\{0\}}, \ldots, \widetilde{v}^{\{k+d-1\}}$ to remove uninteresting zero vectors.

The case of any k vectors as input: An example an algorithm to obtain a basis with (Prop 1) and (Prop 2)

Solution:

- We will use those vectors from $\widetilde{v}^{\{0\}}, \ldots, \widetilde{v}^{\{k-1\}}$, which did not become zero vectors, for the first part.
- We will use those vectors from $\widetilde{v}^{\{k\}}, \ldots, \widetilde{v}^{\{k+d-1\}}$, which did not become zero vectors, for the second part.

They are d orthogonal vectors, and therefore they must be a basis of the vector space $\mathbb{R}^d$ as desired.