Linear Algebra Summary

Apr 2, 2021

Systems of Linear Equations

Linear equation

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

- a's are the **coefficients** also called **weights**
- b is the **constant**

Linear system

$$\begin{gathered}a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = c\end{gathered}$$

- a **solution** is a list $(s_1, s_2, \cdots, s_n)$, also called **s**, where s is usually represented as a vector, since all the x's can also be represented as a vector **x**.

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} \quad s = \begin{pmatrix} s_1 \\ s_2 \\ \cdots \\ s_n \end{pmatrix}$$

- Substituing s for x makes the equation true.
- Two linear systems are **equivalent** if they have the same **solution set**.
- The **solution set** is the set that contains all valid solution vectors.
- The solution set of a system of linear equation has:
  - no solution
  - exactly one solution
  - infinitely many solutions.
- A system is **consistent** if it has 1 or ∞ solutions.
- A system is **inconsistent** if has no solution.

Matrix Notation

We can express the system below as a matrix

$$\begin{gathered}a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = c \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = k \\ = \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & | & b \\ a_{21} & a_{22} & \dots & a_{2n} & | & c \\ \vdots & \vdots & \ddots & \vdots & | & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & | & k \end{bmatrix}\end{gathered}$$

- **coefficient matrix**: Matrix representation that expresses **only** the **coefficients**
- **augmented matrix**: Matrix representation that includes the **constants**, sometimes separated by a vertical line
- The size of a matrix is expressed as **rows x columns** (m x n).

(Reduced) Echelon Matrix

Echelon Matrix

$$\left[\begin{array}{cccc|c} \blacksquare & * & \dots & * & * \\ 0 & \blacksquare & \dots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \blacksquare & * \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Reduced Echelon Matrix

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

- ■ represents any non-zero number and ***** represents any number.
- ■ identify the **pivot** positions, also called **leading entries**.

Echelon form requirements:

- All-zero rows must be placed at the bottom
- Leading entries must be displayed as “downstairs”
  - It is fine if there’s a “gap”/”missing leading entry”
  - In such occasion we find a non-pivot column
- ■ only has zeros below

Reduced echelon form requirements:

- The **leading entries** are 1’s
- The 1’s only have zero’s below **and** above.
- While there are infinity ways to express an echelon matrix (with different scale), there’s only 1 reduced echelon form.
- We call U the reduced echelon form of matrix A

Solving a system

- Write the augmented matrix of the system.
- Bring matrix to the reduced echelon form with row operations.
- Provide the parametric description of the solution set, declare that the solution set is empty or give the vector notation of the unique solution.

Row operations:

1. **Replacement**: Add a scaled row to another row
2. **Interchange**: Swap rows
3. **Scaling**: Scale row by a non-zero constant

Row Reduction Algorithm

Forward phase

1. Start at the first non-zero column
2. Swap rows if needed to keep the pivot entry at the top.
   1. Choose a row whose leading entry is already a 1 or scale it into a 1 leading entry.
3. Work your way down with row operations so that there are only 0s under the pivot entry.
4. Move onto the next column and repeat the same process.

Back-ward phase

1. Start at the bottom right entry
2. Work your way up with row operations and try to remove all zeros above.

Variables

The row reduction algorithm will not always produce a "clean" reduced echelon form where each row has a pivot.

Basic variables

- Variables that correspond to pivots ■

Free variables

- Variables that correspond to non-pivot coeffcients *

Example:

$$\begin{bmatrix} 1 & 0 & -5 & \mid & 1 \\ 0 & 1 & 1 & \mid & 4 \\ 0 & 0 & 0 & \mid & 0 \end{bmatrix}$$

Corresponds to:

$$\begin{aligned} x_1 \qquad - 5x_3 &= 1 \\ x_2 + x_3 \quad &= 4 \\ 0 &= 0 \end{aligned}$$

Which equals the parametric description (of the solution set) below...

Parametric description of solution sets

...example follow up:

$$\begin{cases} x_1 = 1 + 5x_3 \\ x_2 = 4 - x_3 \\ x_3 \; is \; free \end{cases}$$

x_3 is free means that you are free to choose any value for x_3 (also called **paramater**) and each different choice of x_3 determines a (different) solution of the system.

Solving a system amounts to finding a parametric description of the solution set (such as the system of equations above or such as the Parametric Vector Equation) or determining that the solution set is empty:

- If a system contains an equation of the form 0 = b, where be is not zero, then it is inconsistent, else, consistent.
- If the system is consistent and has free variables, then it has ∞ solutions.
- If the system is consistent and has no free variables, then it has 1 solution.
- If a system is inconsistent, the solution set is empty, even if it has free variables.

Spans and vector equations

Vector Equations

- A matrix with only 1 column is a **column vector** or a **vector**
  - A **vector** stands for an "ordered list of numbers".
  - The vector whose entries are all zero is called the **zero vector**, denoted by **0**
- Vector sum:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \end{bmatrix}$$

- Vector scaling:

$$a \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a \cdot u_1 \\ a \cdot u_2 \end{bmatrix}$$

- The set of all vectors with 2 **entries** (“vector rows”/numbers) is expressed as $\mathbb{R}^2$
  - This represents all the vectors within a 2D/plane space
  - Two vectors in $\mathbb{R}^2$ are equal iff all entries are equal and in the same order.
  - The first entry is assigned to the x-coordinate and the second entry to the y-coordinate.
- $\mathbb{R}^3$ represents the set of all vectors in a 3D space
  - 1D and 2D vectors are not implicitly allowed as the vectors in $\mathbb{R}^3$ must have 3 coordinates (x,y,z)
  - However 1Ds may appear as (x,0,0) and 2Ds as (x,y,0) (but with the vertical notation)

Linear Combinations

Let the vector **b** be defined by

$$b = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots$$

- ** b** is a linear combination of ** a_1**, ** a_2... with **weights** (scalars) x_1, x_2, ...
  - observe that while the columns are ** a_1**, ** a_2..., when we combine rows with columns, the row index is placed before the column index.
- The vector equation above has the same solution set as the linear system whose augmented matrix is:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & | & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & | & b_2 \\ a_{31} & a_{32} & \cdots & a_{3n} & | & b_3 \\ \vdots & \vdots & \cdots & \vdots & | & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & | & b_{mn} \end{bmatrix}$$

- **b** can only be generated by a linear combination of **a** iff the augmented matrix has a solution
- The set of all possible linear combinations of a list of vectors a_1, a_2, ... a_n is denoted as Span{ a_1, a_2, ... a_n}
  - That is, all the b’s such that

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots = b$$

- yields an augmented matrix with solution.

A**x** = **b**

A**x**

- This represents the multiplication of a matrix A by an ordered list of numbers in x.
  - This is defined iff the number of entries in x is the same as the number of columns in A.

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots$$

- Ax = b has the same solution set as the vector equation:

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots = b$$

- which has the same solution set as the augmented matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & | & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & | & b_2 \\ a_{31} & a_{32} & \cdots & a_{3n} & | & b_3 \\ \vdots & \vdots & \cdots & \vdots & | & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & | & b_m \end{bmatrix}$$

- Iff A**x** = **b** has a solution, then we can say that **b** is a linear combination of the columns of A (with x as the column weights).
  - This make sense since A**x** = **b** is just the augmented matrix of a linear system where replacing the **x** coefficients for the solution **s** give you the linear combination on the left side of the equation such as it matches the right side vector (**b**) making the equation true.

All true XOR All false

Let A be an n x m matrix, these statements are either all true or all false.

- A**x** = **b** has a solution for all **b**'s in $\mathbb{R}^n$
- All **b**'s in $\mathbb{R}^n$ are a linear combination of A
- Span{ $a_1, a_2, \cdots, a_m$} = $\mathbb{R}^n$
- A has a pivot position in every row

Solution sets and linear independence

Linear dependence

- A set of two vectors { a_1, a_2} is linearly dependent **iff** at least one of the vectors is a multiple of the other.
- A set of 2 or more vectors is linearly dependent **iff** at least one of the vectors is within the span of another vector(s) other than itself. That is, if a vector can be expressed as a linear combination of other vectors in the set.
- A set of 1 vector is always independent except for the **0** vector.
  - This is because $x_1 0 = 0$ has ∞ nontrivial solutions.
- If a set of vectors has more vectors than vector entries, then there will be at least a vector within the span of another vector(s), making the set linear dependent.
  - If a matrix has more columns than rows, then there will be free variables, therefore making the matrix dependent.
- If a set contains the zero vector, the set is linearly dependent
  - If a matrix has an all-zeros column, the matrix columns are linearly dependent.

Homogeneous Linear Systems

- Systems that have the form A**x** = **0**

- Always have at least x = **0** solution (aka **trivial solution**)
- **Nontrivial solution** is a **non-zero vector x** that satisfies A**x** = **0**
- **Ax = 0 has a nontrivial solution iff the system has at least one free variable**
    - You can then use there an arbitrary non-zero number in the solution which makes $x \neq 0$
    - Each of these infinityely many nontrivial solutions of **Ax** = **0** are called **linear dependence relation** among a_1, a_2, ..., a_n

Solution sets and dependence

- The solution set of an homogenous system (A**x**=**0**) can always be expressed as:
    - If it has only trivial, the solution set equals Span{**0**}
        - Which is the (0,...,0) coordinate
        - Which also means that the column vectors of A are **linear independent**
    - If it has nontrivial, the solution set equals Span{ $v_1 \cdots v_n$} (which is not necessarily $\mathbb{R}^n$, think of v_1[1,0,0], v_2[0,1,0], v_3[1,1,0] -assume vertical notation-, it spans a plane in $\mathbb{R}^3$, not the whole $\mathbb{R}^3$)
        - Only 1 free variable Span{ v_1} means the solution is a line through the origin. (i.e. the z-axis in the 3D vectors above)
        - 2 free variables Span{ v_1, v_2} can yield a plane (or a line if v_1 and v_2 are **linear dependent**) through the origin. (i.e. the x-y plane)
        - It also means that there is a parametric Vector Equation where there exists weights other than 0 for x that make A**x**=**0** true
- Therefore, when they ask you, is { $a_1, a_2, \cdots, a_n$} a dependent or an independent set of vectors, they're actually asking...
    - **Dependent: Does Ax=0 have at least a nontrivial solution?**
        - Check if it has at least 1 **free variable.**
    - **Independent: Does Ax=0 have only the trivial x=0 solution?**
        - Check if there are **pivots in every row of the echelon reduced matrix.**

Parametric Vector Equation

The parametric description of a (homogenous) solution set can be expressed in a single equation:

$$x = x_1 v_1 + x_2 v_2 + x_3 v_3 \cdots$$

in CSE1205 $x = s v_1 + t v_2 + u v_3$

Nonhomogeneous Linear Systems

When these systems have $Ax \neq 0$ the system is translated by a vector constant **p** from the origin:

$$x = p + x_1 v_1 + x_2 v_2 + x_3 v_3 \cdots$$

- The vector ** p** itself is just one **particular** solution of A**x** = **b** corresponding to $x_1 \cdots x_n = 0$
    - xv is the **general** solution of A**x**=**0**
- When there is only 1 parameter vector (v) this equation is also called "**the equation of the line through p parallel to v**"
- The solution set of A**x** = **b** is:
    - empty if it has no solution
    - a translate of the (parametric) solution for A**x**=**0**
- The parametric system of equations of the previous example:

$$\begin{cases} x_1 = 1 + 5x_3 \\ x_2 = 4 - x_3 \\ x_3\ is\ free\ (x_3 = 0 + x_3) \end{cases}$$

Is represented as the parametric vector equation:

$$x = \begin{bmatrix} 1 \\ 4 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 5 \\ -1 \\ 1 \end{bmatrix}$$

Linear transformations

All matrix transformations are **linear** functions, but not al linear functions are matrix transformations (i.e f(x)=3x+5). A **matrix** linear transformation T must have the same properties as a Matrix-Vector Product A**x**:

$$(A(\mathbf{u}+\mathbf{v}) = A\mathbf{u} + A\mathbf{v}) \leftrightarrow (T(\mathbf{u}+\mathbf{v}) = T\mathbf{u} + T\mathbf{v})$$

$$(A(c\mathbf{u}) = c(A\mathbf{u})) \leftrightarrow (T(c\mathbf{u}) = c(T\mathbf{u}))$$

For an n x m matrix, the columns are vectors in $R^n \; or \; R^{rows}$ A**x**=**b** would only be defined when the entries in x are the same as the number of columns (m), therefore x is in R^m, but the output b has the same number of entries as A has rows, therefore in R^n

The correspondence from **x** to **Ax** is a **function** or a **transformation** or a **mapping** from one set of vectors in R^m to another in R^n Formally denoted as $f : R^m \rightarrow R^n$, where 'f' just happens to be the arbitrary name that we gave to the function (or transformation, mapping, or even relation, see CSE1300).

The "dimension" of the **inputs** is called the **domain** of 'f', in this case R^m, and the "dimension" of the **output** (formally kown as "image" or vagely written as *f(x)*) is called the **co-domain**, in this case R^n However, this does not necessarily mean that all "dimensions" within the co-domain are "used", as a possible transformation to a co-domain in R^3 could deliberatly have z = 0 for all outputs. The **range** of f (the function/transformation/mapping), is the **set of all outputs**, which happens to be the Span of each of the vector columns of A.

In the example used where z is always 0, we would have a plane as a **range**, but still inside the 3D dimension (it would still have 3 coordinates, even if the 3rd is the same for all outputs). Since we know by the definition of this particular function that the co-domain is in the 3D dimension, we can already deduce that in the case of a $f(x) = A\mathbf{x}$ function, the column vectors of A have to be in the same R^n as the co-domain, R^3 in this case.

Matrix transformations

In this context, $T(x)$ is reserved for $T(x) = A\mathbf{x}$ This matrix transformation is also denoted as $x \mapsto A\mathbf{x}$

When the transformation maps an input to an output in the same dimension, it is called a **shear transformation**:

$$Let \; u = \begin{bmatrix} 0 \\ 2 \end{bmatrix} and \; A = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$$

$$Then \; T(u) = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

When a transformation goes from a higher dimension to a lower one we can say that $x \mapsto A\mathbf{x}$ **projects** points in R^k onto R^{k-1}

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$$

The Matrix of a Linear transformation

Identity matrix

The identity matrix with size n x n, also denoted as I_n, is the matrix of a T(x) transformation such as T(x) = x. The identity matrix returns the input in the same form since each of it’s columns (n) contains only a 1 in the nth entry:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Let\ T(x) = I_2\mathbf{x}$$

$$T(\begin{bmatrix} 5 \\ 3 \end{bmatrix}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$= 5 \begin{bmatrix} 1 \\ 0 \end{bmatrix} 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \mathbf{x}$$

The columns of I_n are formally referred to as $\mathbf{e_n}$ For a 2x2 A matrix we may express the first and second column vectors as $\vec{i}$ and $\vec{j}$

In the same way that in a ‘f(x) = ax + b’ linear function we can draw the line from just 2 points, Since T(x) is also a linear function, that means that knowing the output of 2 inputs is enough information to deduce the matrix used for the Matrix-vector product. The matrix used for the Matrix-vector product is called the **standard matrix for the linear transformation**.

2D Transformations

To easily see what the 2x2 A matrix does to a vector, analyze the difference between the default values for $\vec{i}$ and $\vec{j}$, vs their new values defined by A. The default values are:

$$\vec{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} and\ \vec{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \uparrow \rightarrow$$

90° rotation (counter clockwise)

To do so the green arrow would have to point to (0,1) and the red arrow would have to point to (-1,0). The new values for $\vec{i}$ and $\vec{j}$ would be :

$$\vec{i} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} and\ \vec{j} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

Resulting in:

$$T(x) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A rotation (in rad)

When prompted to calculate the standard matrix for a rotation by α rad, $\vec{i}$ previous coordinates (1,0) will change in the same fashion as the unit circle:

$$\vec{i} = \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \end{bmatrix}$$

The changes for $\vec{j}$ will be the same, but with a $\frac{1}{2}\pi$ offset, as $\vec{j}$ was already 90° counterclockwise from $\vec{i}$

$$\vec{j} = \begin{bmatrix} cos(\alpha + \frac{1}{2}\pi) \\ sin(\alpha + \frac{1}{2}\pi) \end{bmatrix} = \begin{bmatrix} sin(\frac{1}{2}\pi - \alpha - \frac{1}{2}\pi) \\ cos(\frac{1}{2}\pi - \alpha - \frac{1}{2}\pi) \end{bmatrix} = \begin{bmatrix} sin(-\alpha) \\ cos(-\alpha) \end{bmatrix} = \begin{bmatrix} -sin(\alpha) \\ cos(\alpha) \end{bmatrix}$$

Reflection through the x-axis

Here $\vec{i}$ and $\vec{j}$ would have to go from ↑→ to ↓→ That means that only $\vec{j}$ is changed:

$$\vec{j}_{old} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \longrightarrow \vec{j}_{new} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \longrightarrow A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Reflection through the y-axis

Here $\vec{i}$ and $\vec{j}$ would have to go from ↑→ to ←↑ That means that only $\vec{i}$ is changed:

$$\vec{i}_{old} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \longrightarrow \vec{i}_{new} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \longrightarrow A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Reflection through the line y=x

This is also known as the inverse of a function. We simply have to **swipe ROWS**, resulting in:

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

This could also be achieved with:

- rotation of 90° counter clockwise followed by a reflection in the y-axis
- or, reflextion in the x-axis followed by rotation of 90° counter clockwise

Reflection through the origin

This is simply a 180° rotation, or alternatively scale both $\vec{i}$ and $\vec{j}$ by -1, resulting in:

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

Reflection through the line y = -x

This combines a reflection through the origin with an inverse.

$$Inverse = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \longrightarrow Reflection_{origin} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

Horizontal contraction and expansion

$$A = \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$$

If k > 1, then the image is stretched out. If 0 < k < 1, then the image is squeezed. If k < 0 the image is mirrored on the y-axis.

Vertical contraction and expansion

$$A = \begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix}$$

If k > 1, then the image is stretched out. If 0 < k < 1, then the image is squeezed. If k < 0 the image is mirrored on the x-axis.

Horizontal shear

$$A = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

Here $\vec{j}$ is tranformed from $\uparrow$ to $\nearrow$ with k slope.

Which means that as we move up, the image is stretched horizontally by k units.

Vertical shear

$$A = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$$

Here $\vec{i}$ is tranformed from $\rightarrow$ to $\nearrow$ with k slope.

Which means that as we move right, the image is stretched vertically by k units.

I and j are dependent

If both $\vec{i}$ and $\vec{j}$ are dependent (they can be expressed as a linear combination of the other, in this case as a scalar of the other), such as in the matrix with all 1's, we would endup with $\nearrow\nearrow$ It means that both $\vec{i}$ and $\vec{j}$ "collapse" into each other making the image of the transformation get completly squeezed in the line that goes through both $\vec{i}$ and $\vec{j}$

Projection onto the x-axis

The transformed image only has x-values (y remains constant 0).

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

Projection onto the y-axis

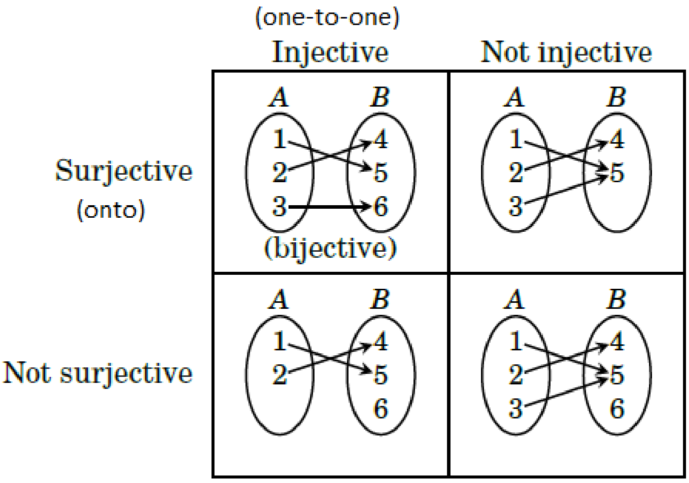
The transformed image only has y-values (x remains constant 0).

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

Surjective (onto) and injective (one-to-one) mappings

- A mapping $T : A \rightarrow B$ is **surjective** (A is **onto** B) iff Range = Codomain.
  - A has a **pivot position in every row**
    - A**x** = **b** has a solution for all **b**'s in $\mathbb{R}^n$
      - All **b**'s in $\mathbb{R}^n$ are a linear combination of A
    - The **columns of A span** R^n
- A mapping $T : A \rightarrow B$ is **injective** (A is **one-to-one** with B) iff every input gets a unique output.
  - The **columns of A are linearly independent**
    - **T(x) is injective (onte-to-one) iff T(X) = 0 only has the trivial solution** (not being injective doesn't have to make you surjective and viceversa)
- A mapping is bijective if it is both, surjective and injective.
- $T : R^n \rightarrow R^{n+1}$ is at most injective (if columns are independent)
- $T : R^n \rightarrow R^{n-1}$ is at most surjective (if every row has a pivot)
- $T : R^n \rightarrow R^n$ is is either bijective or neither surjective nor injective

Hint from the Book of Proof (Richard Hammack):



Composition of 2 transformations

Where A_1 and A_2 are the standard matrixes for the the 1st and the 2nd transformations respectively:

$$T(\mathbf{x}) = A_2\Big(A_1\begin{bmatrix} x \\ y \end{bmatrix}\Big)$$

Which results in a matrix multiplication, explained in the chapter below.

Matrix operations

Matrix rules

- **ORDER MATTERS:** $A_2 \cdot A_1 \neq A_1 \cdot A_2$
- cancellation laws do not apply in matrix algebra
- **associative law**: $A(BC) = (AB)C$, becuase the order is the same! (always right to left).
- **distributive law**:
  - A(B+C) = AB + AC (keep A on the side where it was)
  - (B+C)A = BA + CA
- r(AB) = (rA)B = A(rB) for any scalar r
- $I_n A = A I_n = A$
- **A TRANSLATION IS NOT A LINEAR TRANSFORMATION**. In a transformation, the origin is always in the same spot. In a translation it is not.
- $A^k = A \cdot A \ldots A$ for k times.
- $A^0 = I_n$

Matrix multiplication

From the example above where the composition of 2 transformation was

$$A_2\Big(A_1\begin{bmatrix} x \\ y \end{bmatrix}\Big)$$

in which we can clearly see that first we compute the things inside the brackets, and then the things outside, matrix multiplications come from this concept, and **matrix products are therefore read from right to left**.

Following up the example of 2 transformations. The matrix of the first transformation A_1 contains 2 columns, the updated versions of $\vec{i}$ and $\vec{j}$

$$A_1 = \begin{bmatrix} \vec{i}_{A_1} & \vec{j}_{A_1} \end{bmatrix}$$

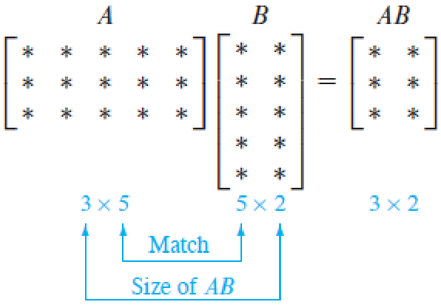
Then, to identify the new $\vec{i}$ and $\vec{j}$ for A_2, we'd have to apply a vector multiplication of $\vec{i}_{A_1}$ and $\vec{j}_{A_1}$ with A_2 respectively. It's easier to think about the transformation by focusing on just one vector at a time, so we apply 2 sequent transformations for $\vec{i}$ and $\vec{j}$ respectively, and once we're done, we join $\vec{i}_{A_2}$ and $\vec{j}_{A_2}$ into a single matrix A_2

$$\vec{i}_{A_2 \cdot A_1} = A_2 \cdot \vec{i}_{A_1}$$

$$\vec{j}_{A_2 \cdot A_1} = A_2 \cdot \vec{i}_{j_1}$$

$$A_2 \cdot A_1 = \begin{bmatrix} \vec{i}_{A_2 \cdot A_1} & \vec{j}_{A_2 \cdot A_1} \end{bmatrix}$$

Following the properties of the matrix-vector product, the matrix-matrix product (which is nothing but a transformation of a transformation), must have the number of columns in matrix A be equal to the number of rows in matrix B in AB, and AB size is $rows_A \times columns_B$



Row-column multiplication

There's a shortcut formula to get the value of AB's a_{ij} where i is also A's i row and j is also B's j column: Take A's i row and transpose it to a vector, then calculate the dot product with B's j column:

$$AB = \begin{bmatrix} 2 & 3 \\ 1 & -5 \end{bmatrix} \begin{bmatrix} 4 & 3 & 6 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} \square & \square & 2(6)+3(3) \\ \square & \square & \square \end{bmatrix} = \begin{bmatrix} \square & \square & 21 \\ \square & \square & \square \end{bmatrix}$$

Zero matrix

Expressed as 0, it's the equivalent of a n x m (size assumed by context), with all zeros.

Square matrix

- number of rows = number of columns
- Therefore defined as n x n

Identity matrix

Expressed as I_n It is a n x n **diagonal matrix** (all of the diagonal matrix are n x n), and as any diagonal matrix, it is a zero matrix besides for the **(main) diaogonal entries**, these happen to be all 1's and the property that it has is $A \cdot I_n = A$

Lower/upper triangular matrix

All entries above/below the main diagonal are zeros.

Matrix sum

Sum the columns, only defined when both matrixes have the exact same dimensions.

Scalar multiplication

Scale every column. The scalar is only defined when it comes to the left of the matrix. The right side of the matrix should be reserved for matrixes or vectors (which are just 1 column matrixes).

Transposing

You sawp the columns for the rows (just like in excel), it's **not** a 90° rotation. There are some properties:

- $(A^T)^T = A$
- $(A+B)^T = A^T + B^T$
- For any scalar r $(rA)^T = rA^T$
- $(AB)^T = B^T A^T$
- If A is invertible: $T^{-1}(x) = A^{-1}x$

Inverse matrices

Not all matrices have an inverse. The inverse of A is A^{-1} and it should hold that:

$A^{-1}A = I$ and $AA^{-1} = I$

A matrix that is not invertible is a **singular matrix** suchas the zero matrix, and an invertible matrix is called **nonsigular matrix**.

Example of a **nonsingular matrix** would be a $T : R^2 \to R^2$, suchas a 90° rotation **counterclockwise**:

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

The inverse of A would be to rotate 90° **clockwise**:

$$A^{-1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

If we apply both transformations, intution says that $\vec{i}$ and $\vec{j}$ should return to their default values, and hence $A^{-1}A = I$ and $AA^{-1} = I$

$$A^{-1}A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

The same would apply to $AA^{-1} = I$ This was possible since A has independent columns ($\vec{i}$ and $\vec{j}$ are not a linear combination of each other), and therefore there is only 1 solution to T(X) = 0. If we had multiple solutions for T(x) = 0, that wolud mean that T(X) is no longer a one-to-one mapping (injective), and when we try to come up with an inverse function, an input would have multiple possible outputs, which is not allowed in a function.

Furthermore, if we apply the zero matrix transformation, it is impossible to restore the default values of $\vec{i}$ and $\vec{j}$ (for the identity matrix) when all you have is all zeros... So to analyze if the matrix is invertible, you can check if the domain and co-domain remain the same, if the matrix is dependent, and to visualize the transformation and think how to restore $\vec{i}$ and $\vec{j}$

There's also a formula to find the inverse of a 2x2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \wedge ad - bc \neq 0 \to A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$ad - bc = 0 \to A \; is \; not \; invertible$$

$ad - bc$ is formally known as the **determinant (det A) for a 2 by 2 matrix A**, which for a $T : R^m \to R^n$ is the change in volume (3D), space (2D)... between I and A

A^{-1} exists iff the $detA \neq 0$

Some properties for invertible matrices A nd B:

- $(AB)^{-1} = B^{-1}A^{-1}$
- $(A^T)^{-1} = (A^{-1})^T$

- if A is an invertible n x n matrix, then for reach b in R^n the equation Ax = b has the unique solution $x = A^{-1}b$

Invert A column by column

If A is invertible, let e_n be the nth column of I Then for each of the rows of A (and columns of the inverse a_n^{-1}) find:

$$Aa_n^{-1} = e_n$$

Which in turn can be translated to an augmented Ax=b echelon matrix with the coefficent matrix of A and with b the nth column of I The resulting vector is the nth column of A^{-1}

Invert A as a whole

If A is invertible, row reduce the augmented matrix $[A \ \ I]$ until it becomes $[I \ \ A^{-1}]$

Invertible matrix theorem

For a square n x n A matrix, these are either all true or all false:

- A is an invertible matrix
- A is row equivalent to the n x n identity matrix
- A has n pivot positions
- A**x** = **0** has only the trivial solution
- The columns of A are independenet
- T(x) is one-to-one
- Ax = b has at least one solution for each b in R^n
- The columns of A span R^n
- T(x) is $T : R^n \to R^n$
- The transpose of A is also invertible

Invert 2 by 2 fast

$$Let\ A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

This all true or all false apply only to square matrices.

- For a n x n matrix biger than 2 x 2, the easiest way to check if it's invertible is to solve for A**x** = **0** and check if there's only the trivial solution (if there are infinite solutions then it is not invertible).

Subspaces of R^n

These sets of vectors provide useful information about the equation A**x**=**b**.

A subspace of $\mathcal{R}^n$ is any set H in $\mathcal{R}^n$ that has:

- zero vector $\in H$
- all linear combinations of its initial vectors are $\in H$

Column space (Col A)

- The **column space** of a matrix A is the set of all linear combinations of its vector columns, denoted as **Col A**
- The **pivot columns** of a matrix A form a basis for the column space of A.
  - YOU NEED THE ORIGINAL PIVOT COLUMNS OF A, NOT THE COLUMNS OF THE ECHELON FORM.
- The column space of an n x m matrix is a subspace of R^n

Null space (Nul A)

- The **null space** of a matrix A is the set of all solutions to the equation Ax = 0, denoted as **Null A**.
- The null space of an n x m matrix is a subspace of R^m (in Ax=B the x vector has m entries)

Zero subspace

Set {**0**}

Basis subspace

- The smallest possible spanning set of linearly independent vectors that span H.

Coodinate systems

The goal of a using the basis for a subspace H (rather than a Span set) is to have a minimal set of vectors that operate as "i, j… hat", from which we can define any point laying on H, as a linear combination of the basis vectors. Usually the basis for H is defined as $\mathcal{B} = \{b_1, \ldots, b_p\}$

The **coordinate vector of x (relative to $\mathcal{B}$) or the B-coordinate vector of x** is defined as:

$$[x]_{\mathcal{B}} = \begin{bmatrix} c_1 \\ \vdots \\ c_p \end{bmatrix}$$

Often H is a plane in R^3, the B-coordinate vectors are still R^3 However, the transformation $x \mapsto [x]_B$ is called "isomorphic" as H is isomorphic to R^2

$Let\ \mathcal{B}$ ={ b_1, b_2, b_3}

$$If\ \mathbf{x}\ is\ a\ vector\ in\ H\ with\ [x]_{\mathcal{B}} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$
$$then\ \mathbf{x} = c_1 b_1 + c_2 b_2 + c_3 b_3$$

Dimension

Because H is given with the minimal number of independent vectors, the dimension of a nonzero subspace H, denoted by *dim H*, is the number of vectors in any basis for H. The dimension of the zero subspace {**0**} is defined to be zero.

- Technically the zero subspace has no basis because the zero vector itself forms a linearly dependent set.

- The **rank** of a matrix is te dimension of the column space of A.

If a matrix A has n columns, then some are free variables (dimNul(A)) and some are legit basis (pivots, which add to rank(A)), the sum of the numer of pivots and the number of free variables = n, formally:

$$rank(A) + dim(Nul(A)) = n$$

For an n x n matrix, iff A is invertible, then:

- The columns of A form a basis of R^n
- Col A = R^n
- dim Col A = n
- rank A = n
- Nul A = [**0**]
- dim Nul A = 0

Determinants

- Determinants are closely linked to the eigenvalues.
- The determinant (det A or |A|) is a number associated only with square matrixes
- The square matrix is invertible when the determinant is not zero and single otherwise

Property 1

- The determinant of the identity matrix I is 1.
  - This is because A*I=A (a is scaled by 1)

Property 2

- In a Permutation of A (that is an exchange of rows), if the number of exchanges is odd, the determinant is changed by scalar: -1
- In a Permutation of A, if the number of exchanges is even: 1

The determinant of a 2 by 2 matrix is:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Property 3a

- If we multiply the first row of a matrix A by a scalar t, the determinant becomes t * det A

$$\begin{vmatrix} ta & tb \\ c & d \end{vmatrix} = t \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

- Similarly, we can factor out a scalar t from a row

Property 3b

$$\begin{vmatrix} a+a' & b+b' \\ c & d \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} a' & b' \\ c & d \end{vmatrix}$$

- If we add a’ to all rows these can be row reduced. The property only applies when 1 row is incremented.
- If a’ and b’ are zero, then we would have det A = det A + 0.
- $det 2A = 2^n det A$

$$Let\ A\ be\ 2 \times 2 \rightarrow |kA| = \left| k \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right| = \begin{vmatrix} ka & kb \\ kc & kd \end{vmatrix} = k^2ad - k^2bc = k^2(ad - bc) = k^2 det A$$

Propery 4

- If two rows are equal, the determinant is zero.
  - We already know that the Rank is less than n, and therefore it is not invertible. So the determinant is zero.
  - But this also comes from property 2
    - If we have 2 equal rows we can switch them, which should make the determinant of the opposite sign. However, the matrix still looks the same. The only number that is equal to it’s opposite sign is 0.

Property 5

- Determinant doesnt change from row addition of a multiple of another row.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = \begin{vmatrix} a & b \\ c - la & d - lb \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} a & b \\ -la & -lb \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} - l \begin{vmatrix} a & b \\ a & b \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

- We can use this property to achieve triangular matrices whose determinant is calculated by multiplying the elements in the main diagional (technically we are using property 3 to factor out each of the scalars multiplying each of the rows of the identity matrix).

Property 6

- Row of zeros $\rightarrow$ det A = 0

Property 7

- The determinant of an echelon form matrix is equal to the product of the diagonal (product of pivots), and the sign depends on the number of row exchanges we made to get into the desired triangular form (+ if even, - if odd).

$$det\ U = \begin{vmatrix} d_1 & * & \dots & * \\ 0 & d_2 & \vdots & * \\ 0 & 0 & \ddots & * \\ 0 & 0 & \dots & d_n \end{vmatrix} = d_1 d_2 \cdots d_n \begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \vdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{vmatrix} = d_1 d_2 \cdots d_n$$

Property 8

- det A = 0 when A is singular (not invertible)
- det A != 0 when A is invertible

Det A formula

- Bring matrix to echelon form and multiply the pivots

Example:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ c - \frac{c}{a}a & d - \frac{c}{a}b \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ 0 & d - \frac{c}{a}b \end{bmatrix} \rightarrow detA = a(d - \frac{c}{a}b) = ad - cb$$

- The reason we don't care about the numbers above the pivot positions is because these can be turned into zeros with row reduction (without swapping rows).

Property 9

- det(AB) = det A * det B
- det $A^{-1} = 1/detA$
- $det(A^2) = (detA)^2$
- $det2A = 2^n detA$ (property 3)

Property 10

- $detA^T = detA$
  - Therefore all the things that applied to rows also apply to columns

Algorithm for determinant 2 by 2

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = \begin{vmatrix} a & 0 \\ c & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & d \end{vmatrix} =$$

$$\begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix} + \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix} + \begin{vmatrix} 0 & b \\ 0 & d \end{vmatrix} =$$

$$0 + ad + \begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix} + 0 =$$

$$ad - bc$$

Algorithm for determinant 3 by 3

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = \begin{vmatrix} a & 0 & 0 \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & 0 & c \\ d & e & f \\ g & h & i \end{vmatrix} =$$

$$\begin{vmatrix} a & 0 & 0 \\ d & 0 & 0 \\ g & h & i \end{vmatrix} + \begin{vmatrix} a & 0 & 0 \\ 0 & e & 0 \\ g & h & i \end{vmatrix} + \begin{vmatrix} a & 0 & 0 \\ 0 & 0 & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ d & 0 & 0 \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ 0 & e & 0 \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ 0 & 0 & f \\ g & h & i \end{vmatrix} + \dots =$$

$$\dots = (3^3\ matrices) =$$

$$\begin{vmatrix} a & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & i \end{vmatrix} + \begin{vmatrix} a & 0 & 0 \\ 0 & 0 & f \\ 0 & h & 0 \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ d & 0 & 0 \\ 0 & 0 & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ 0 & 0 & f \\ g & 0 & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & c \\ d & 0 & 0 \\ 0 & h & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & c \\ 0 & e & 0 \\ g & 0 & 0 \end{vmatrix} =$$

$$aei - afh - bdi + bfg + cdh - ceg$$
$$(3!\ terms)$$

Big formula for determinant of n by n

$$detA = \sum_{n! terms} \pm a_{1\alpha} a_{2\beta} a_{3\gamma} \dots a_{n\omega}$$
$$(\alpha, \beta, \gamma, \dots, \omega) = permutation\ of\ (1, 2, 3, \dots, n)$$

Co-factor algorithm with a 3 by 3

$$aei - afh - bdi + bfg + cdh - ceg =$$

$$a(ei - fh) + b(fg - di) + c(dh - eg)$$

This is one of the many ways you can refractor the 3 by 3 formula.

Graphically, this is the same as picking an arbitrary row (or column, but let’s just stick to rows for now), and calculating the sum of the “available 2 by 2 determinants” times a given element of the row. For the middle element the “available determenintant “wraps around””. See graphical examples below.

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = \begin{matrix} a & - & - \\ | & e & f \\ | & h & i \end{matrix} + \begin{matrix} - & b & - \\ d & | & f \\ g & | & i \end{matrix} + \begin{matrix} - & - & c \\ d & e & | \\ g & h & | \end{matrix} = a(ei - fh) + b(di - fg) * -1 + c(dh - eg)$$

the second term is -1 because we had to flip the columns once. Going back to the ‘Big Formula’, you should have n! terms.

Therefore, co-factor of $a_{ij} = C_{ij} =$

$$\pm det \begin{pmatrix} n - 1\ matrix \\ without\ row_i, col_j \end{pmatrix}$$

The sign of the determinant is determined by the parity of a_{ij}

Plus if i+j is even, minus if odd.

$$\begin{matrix} + & - & + \\ - & + & - \\ + & - & + \end{matrix}$$

Like a n x n chessboard with $a_{11} = +$

Cofactor formula

- for rows (pick constant i):

$$detA = a_{i1}C_{i1} + a_{i2}C_{i2} + \cdots + a_{in}C_{in}$$

- for columns (pick constant j):

$$detA = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj}$$

Cofactor for a 4 by 4

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \rightarrow \begin{pmatrix} + & - & + & - \\ - & + & - & + \\ + & - & + & - \\ - & + & - & + \end{pmatrix}$$

$$\begin{vmatrix} a_{11} & & & \\ & a_{22} & a_{23} & a_{24} \\ & a_{32} & a_{33} & a_{34} \\ & a_{42} & a_{43} & a_{44} \end{vmatrix} + \begin{vmatrix} & a_{12} & & \\ a_{21} & & a_{23} & a_{24} \\ a_{31} & & a_{33} & a_{34} \\ a_{41} & & a_{43} & a_{44} \end{vmatrix} + \begin{vmatrix} & & a_{13} & \\ a_{21} & a_{22} & & a_{24} \\ a_{31} & a_{32} & & a_{34} \\ a_{41} & a_{42} & & a_{44} \end{vmatrix} + \begin{vmatrix} & & & a_{14} \\ a_{21} & a_{22} & a_{23} & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & \end{vmatrix}$$

This can be solved recursively for any n by n matrix. But it would take too much time. It may be more efficient to row reduce a matrix and go for the product of the main diagonal (do not forget the sign changes from row swaps).

Co-factor of 3 by 3

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \rightarrow \begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$$

$$\begin{vmatrix} a_{11} & & \\ & a_{22} & a_{23} \\ & a_{32} & a_{33} \end{vmatrix} + \begin{vmatrix} & a_{12} & \\ a_{21} & & a_{23} \\ a_{31} & & a_{33} \end{vmatrix} + \begin{vmatrix} & & a_{13} \\ a_{21} & a_{22} & \\ a_{31} & a_{32} & \end{vmatrix} =$$

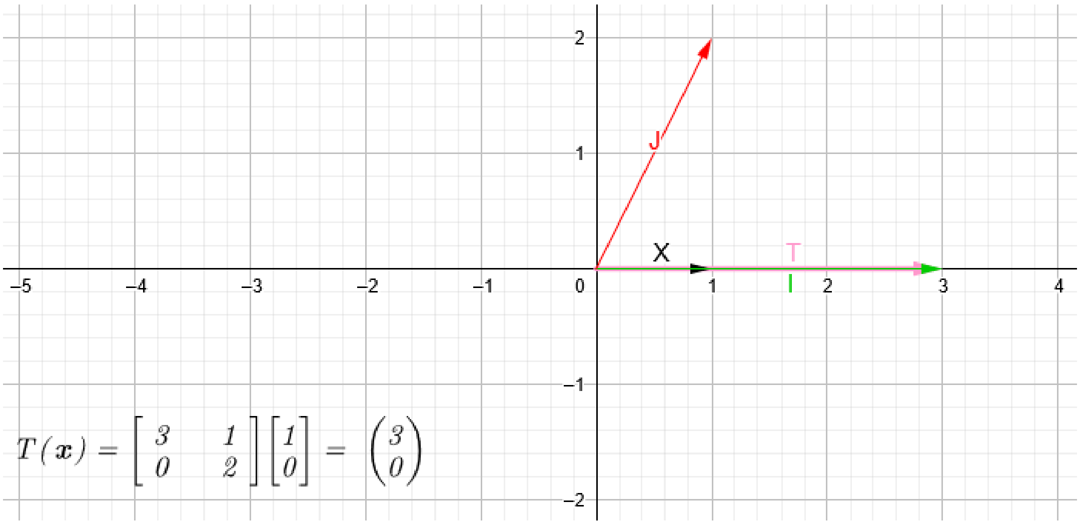
$$a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

Determinant of n by n

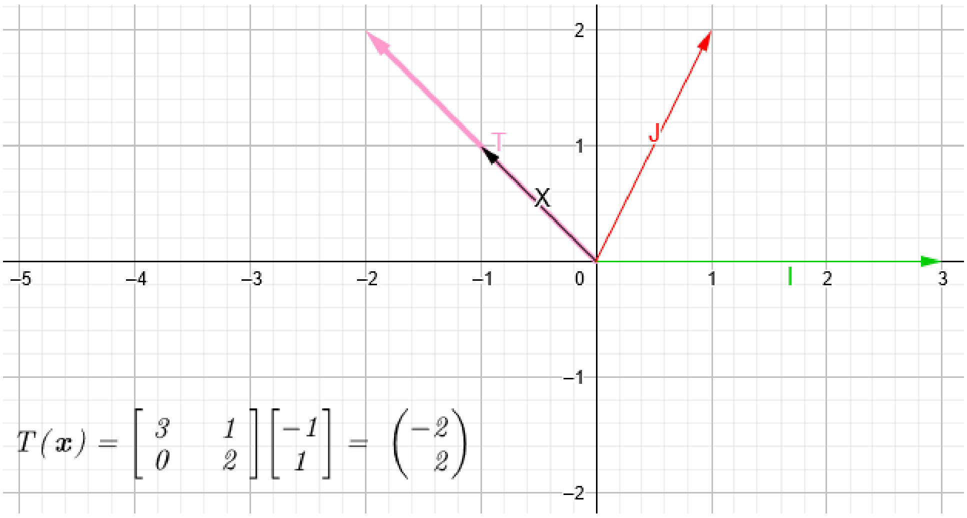
- The strategy is to apply the Cofactor technique by strategly selecting rows or columns with the maximum amount of zeros (to reduce the number of calculations).
  -
    - Because of property 10, you can also aim to create zeros vertically and use the entries of a column as Co-factors
- We can increase the amount of zeros by adding a scaled row to another row, as this operation does not change the sign or value of the determinant.
  - Remember that linear dependence is obvious when two columns or two rows are the same or a column or a row is zero. Which would then make the determinant 0.
- Alternatively, we could also use row reduction to achieve an echelon matrix and calculate the determinant by multiplying all the pivots. Just remember how many row swaps you did (if odd $\rightarrow$ change the sign, else don't).
  - Even if the not reduced echelon form is not unique, the product of the diagonal is!

Eigenvectors and eigenvalues

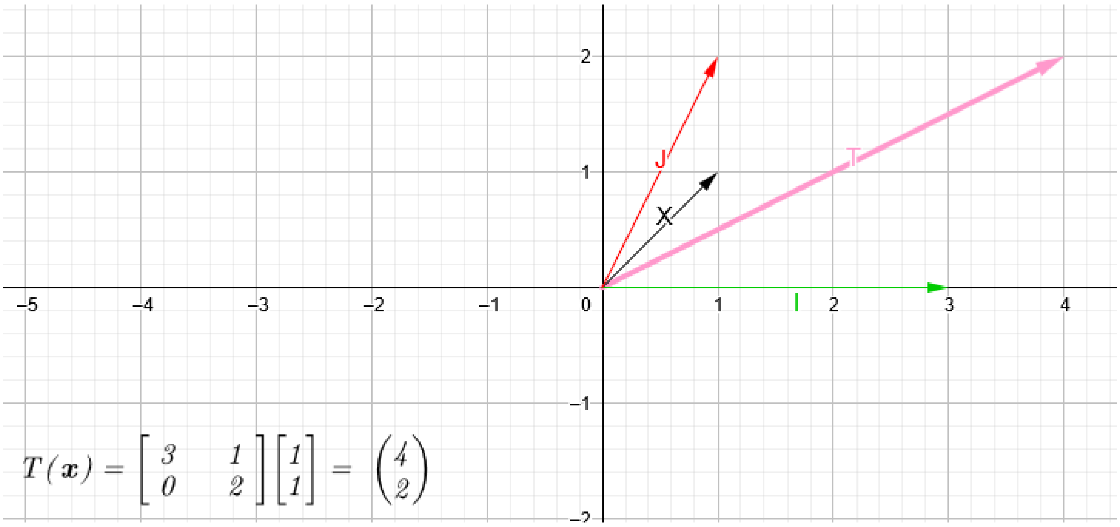
Going back to the linear transformations, we find that for some standard matrices A of a transformation, there are vector inputs such that the output after the transformation is a scalar of themselves (aka, it is within their own span, it's "parallel"), such vector inputs are called eigenvectors, and the scalars are called eigenvalues. For instance, matrix A below has 2 eigenvectors (by definition an eigenvector must have 1 eigenvalue). One of the eigenvectors is X = (1,0), and since T = (3,0) we can see that the eigenvalue is 3, since 3 * X = T.


J
X
T
I
−5 −4 −3 −2 −1 0 1 2 3 4
2 1 −1 −2
$T(x) = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$

The other eigenvector is X = (-1,1), which gives T = (-2,2), therefore the eigenvalue being 2.


T
J
X
I
−5 −4 −3 −2 −1 0 1 2 3
2 1 −1 −2
$T(x) = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$

You can see an example of a non eigenvector below, as X = (1,1) does not produce an output within it's on spawn T = (4,2)


J
T
X
I
−5 −4 −3 −2 −1 0 1 2 3 4
2 1 −1 −2
$T(x) = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$

3D rotation

- In a 3D rotation of an object the eigenvector is the rotation axis, and since we are only rotating the object the eigenvalue would be 1.

Calculate eigenvectors and eigenvalues

According to the definition of an eigenvector and eigenvalue we have $A\mathbf{x} = \lambda\mathbf{x}$ Where A is the standard transformation matrix, $\mathbf{x}$ is the eigenvector and λ is the eigenvalue (scalar).

This is equivalent to $A\mathbf{x} = (\lambda I)\mathbf{x} \longrightarrow A\mathbf{x} - (\lambda I)\mathbf{x} = \mathbf{0}$

$(A - \lambda I)\mathbf{x} = \mathbf{0}$

The main take away from this formula is that besides **v** being the **0** vector (trivial solution), if we want to find another solution (eigenvector), that would mean that more than one input share the same output, which means the transformation is no longer injective. Since it is no longer injective, is no longer bijective, which means $(A - \lambda I)$ is not invertible (is singular) and must have determinant equal to zero.

Step 1 - The characteristic equation

All we need to do is to tweak the value of λ such that $det(A - \lambda I) = 0$ This equation is known as the "characteristic equation" or the "eigenvalue equation".

Step 2 - Null space of A

Now that we know λ, we can actually compute $(A - \lambda I)\mathbf{x} = \mathbf{0}$ and solve it with an augmented matrix. Esentially we are solving for the null space of A.

Example

Using the same matrix A. Let's calculate the eigenvalue and eigenvector.

Step 1

$$det(A - \lambda I) = 0$$

$$det\left(\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix}\right) = 0$$

$$(3-\lambda)(2-\lambda) = 0$$

$$\lambda = 3 \vee \lambda = 2$$

Step 2.1 (lambda = 2)

$$(A - \lambda I)\mathbf{x} = \mathbf{0}$$

$$(A - 2I)\mathbf{x} = \mathbf{0}$$

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

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\mathbf{x} = 0$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$v_1 = -v_2$$
$$v_2 = free$$

$$\mathbf{x} = \mathbf{0} + v_2\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Step 2.2 (lambda = 3)

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

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x} = 0$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$v_1 = free$$
$$v_2 = 0$$

$$\mathbf{x} = \mathbf{0} + v_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Shortcut equation for 2 by 2

$$\lambda^2 - trace\lambda + det(A)$$

Possible eigen values

We see that from the determinant equation we end up with $(a - \lambda)(b - \lambda)$ and sometimes $\lambda^2 = a$ In the first scenario, if a and b are different we get 2 eigenvalues, if they're the same we get 1 eigenvalue (and we would also say that such eigen value has a multiplicity of 2). In the second scenario, if a is positive we get 2 eigenvalues, if a is 0 we get no eigenvalue as by definition 0 cannot be an eigenvalue. If a is negative we get 0 eigenvalues (we actually get an imaginary number, which means we get a rotation for all inputs).

- A single eigenvalue can be used by more than one eigenvector
  - i.e. $A = 2I$ has eigenvalue 2 and all vectors in R^2 are eigenvectors with eigenvalue 2
  - If you compute the eigenvectors from the augmented matrix, you'll see that the augmented matrix only has free variables
- If A matrix has an eigenvalue 0, that means that $Ax = 0x \longrightarrow Ax = 0$ has a nontrivial solution, which means A is not invertible (A is singular), and det A = 0.
  - The eigenvectors with eigenvalue 0 (which is a legit eigenvalue) are the vectors of the Null space of A (the nontrivial vectors that make A**x**=0).
- If the eigenvalues refer only to one eigenvector each, then all eigenvectors are linearly independent
- If two matrices have the same eigenvalues, with the same multiplicities, linked to the same eigenvectors, then these are "similar".
  - "Similar" is not the same as row equivalent
  - Adding kI to a matrix increases the eigenvalues by k but does not change its eigenvectors.
  - Eigenvalues and eigen values are not linear, you can't conclude anything from adding A+B
  - Matrices A and B are similar if there is an invetrible matrix P such that $A = PBP^{-1}$
- A n by n matrix will have n eigen values (in practice some of them can be repeated)
- The sum of the eigenvalues equals the sum of the main diagonal of A (this sum is also called the "trace")
- The determinant is the product of the eigenvalues
- A nonzero determinant does not guarantee an eigenvector
  - The rotation matrix has determinant 1 but eigenvalues are the imaginary number i and -i but no eigenvectors.
- n by n matrix does not guarantee n independent vectors
- The geometric multiplicity of an eigen value is the dimension of the corresponding eigenvector.
- The eigenvalues of a triangular matrix (regardless of whether it is upper or lower zeros) are the entries on its main diagonal
- For A being a n by n matrix, A is invertible iff there is no eigenvalue 0

Eigenspace

The eigen space E_λ is the set of all eigenvectors and the zero vector. In other words, the set of all solutions to the equation:

$$A\mathbf{x} = \lambda\mathbf{x}$$

$$A\mathbf{x} - \lambda\mathbf{x} = 0$$

$$(A - \lambda)\mathbf{x} = 0$$

This is also known as $E_\lambda = Nul(A - \lambda I)$, which is a subspace of R^n

Diagonalization

- A matrix is said to be diagonalizable if it's square, with all 0's except in the main diagonal.
- A matrix A is diagonalizable only if it has n linearly independent eigenvectors.
- A matrix with n distinct eigenvalues has n linearly independent eigenvectors. But you may also get n linearly independent eigenvectors from just 1 shared eigenvalue (i.e $\lambda = 1$ in I).
- The dimension of the eigenspace (the number of linearly indepenent vectors in the set), is less than or equal to the multiplicity of the eigenvalue. So, if an eigenvalue appears twice, it has at most 2 eigenvectors
- If zero is an eigenvalue of A then the matrix A is not invertible, but it can still be diagonazilable.
- The sum of the dimensions of the eigenspaces must be equal to n in order for A to be a diagonaziable matrix.

Similar matrix

- A is similar to B if there exists an invertible matrix P such that $A = PBP^{-1}$
- Similar matrices have same eigenvalues, multiplicities, and eigenvectors.
- A matrix A is diagonazilable if it can be expressed as a similar matrix that is a diagonal matrix.
    - A must have n linearly independent vectors
    - Which form a basis for R^n
    - The algebraic multiplicity of the eigenvalue is equal to the geometric multiplicity of its eigenspace.

Let S be the matrix containing the eigenvectors of A and S^{-1} is the inverse. For that we need n independent eigen vectors.

By the definition of the eigenvector, when you multiply As_1 you're recreating $Ax = \lambda x$, therefore $As_1 = \lambda_1 s_1$ so $AS = \begin{bmatrix} \lambda_1 s_1 & \lambda_2 s_2 & \dots & \lambda_n s_n \end{bmatrix}$

We can factor out the lambdas:

$$AS = \begin{bmatrix} s_1 & s_2 & \dots s_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ & & \ddots & \vdots \\ 0 & 0 & & \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} = S\Lambda$$

Where capital Lambda is the matrix with 0s and the eigenvalues across the main diagonal.

From $AS = S\Lambda$ we also get:

$$S^{-1}AS = \Lambda$$

$$A = S\Lambda S^{-1}$$

If $Ax = \lambda x$ then $A^2x = \lambda Ax = \lambda^2 x$

Similarly, we also have $A^2 = S\Lambda S^{-1}S\Lambda S^{-1} = S\Lambda^2 S^{-1}$ So:

$$A^k = S\Lambda^k S^{-1}$$

The eigenvalues change by the power of k while the egenvectors remain the same.

- If all absolute eigenvalues are smaller than 1, then $A^\infty = 0$ (given that all eigenvalues have multiplicity of 1)
- A is sure to have n independent eigenvectors (and be diagonalizable) if all the eigenvalues are different.
- A diagonal matrix has its eigenvalues sitting infront of you in the main diagonal

Complex eigenvalues and eigenvectors

- Here we allow matrices to be in the $\mathcal{C}^n$ domain, rather than R^n
  - Now eigenvalues can also be complex numbers
- If n by n matrix A has λ eigenvalue and $\mathbf{v}$ eigenvector, then their conjugates also exist
  - The conjugate of a complex number is the same as the original one, but the imaginary part changes its sign, and its associated with the conjugate of lambda.
- Let A be a real 2 x 2 matrix with eigenvalues $a \pm bi$ where $b \neq 0$ and a and b are real numbers. Then there exists an invertible real matrix P such that:

$$A = P \begin{bmatrix} a & -b \\ b & a \end{bmatrix} P^{-1}$$

- The middle matrix with a's and b's is called C.
- Then the eigenvalues of that matrix are $\lambda = a \pm bi$

The matrix P can be constructed as

$$P = [Re\mathbf{v}\ Im\mathbf{v}]$$

Where v is an eigenvector associated to the eigenvalue a - bi.

- The same way that the conjugate of an eigenvalue is another eigenvalue in the matrix, the conjugate of an eigenvector is also another eigenvector
- $r = |\lambda| = \sqrt{a^2 + b^2}$

$$C = r \begin{bmatrix} \frac{a}{r} & -\frac{b}{r} \\ \frac{b}{r} & \frac{a}{r} \end{bmatrix} = r \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

- The angle ϕ is called the argument of $\lambda = a + bi$ (which is arctan(b/a)) and the rotation of the transformation $x \mapsto Cx$ and scaled by $|\lambda|$

Discrete dynamical systems

Recurrence relation to matrix equation

$$\begin{cases} a_k = 2a_{k-1} \\ b_k = 3b_{k-1} \end{cases} \Leftrightarrow \begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} a_{k-1} \\ b_{k-1} \end{bmatrix}$$

$$x_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix} = \left(\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \right)^k \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}$$

- These recurrent relations where $x_k = Ax_{k-1}$ (for all k > 0) are known as discrete dynamical systems
  - Discrete because it refers to different states over time
- x_k is known as a state vector and x_0 is known as the initial state vectors

Coupled system

$$\begin{cases} a_k = 6a_{k-1}\ - 2b_{bk-1} \\ b_k = 6a_{k-1}\ - b_{k-1} \end{cases} \Leftrightarrow \begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ 6 & -1 \end{bmatrix} \begin{bmatrix} a_{k-1} \\ b_{k-1} \end{bmatrix}$$

- Need to diagonalize:

Use $A^k = S\Lambda^k S^{-1}$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

The eigenvectors of S can be scaled such that the columns of S don't have fractions. Get the inverse of S quickly with:

$$Let\ S = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$S^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- A matrix can only be diagonaziable if S (also called P) has n independent columns (eigenvectors). In other words, A is diagonaziable iff its eigenvector matrix is invertible.
- Therefore $x_k = \left(S\Lambda^k S^{-1}\right) x_0$ or, if you know the previous transformation $x_k = Ax_{k-1}$
- This transformation overtime (discrete system) has the impact of:
- The eigenvalues regard the absolute size:
    - attracting vectors towards the origin if all eigenvalues are smaller than 1
    - repelling vectors away from the origin if all eigenvalues are larger than 1
        - except the constant zero trajectory
    - saddle point if some lambdas are bigger than 1 and others smaller than 1
        - Some tend towards the origin
        - Some repell away

Fibonaci Example

0, 1, 2, 3, 5, 8, 13,.. $F_k = ?$

And how fast it grows?

$$F_{k+2} = F_{k+1} + F_k$$

But this is a single equation, not a system. And it's a second order (as there are 2 unkowns).

Trick

We artificially make the single equation into a system, then artificially make the 2 unkowns into a single unkown vector:

$F_k + F_{k+1} = F_{k+2}$

$0 + F_{k+1} = F_{k+1}$

So:

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

And:

$$u_{k+1} = \begin{bmatrix} F_{k+2} \\ F_{k+1} \end{bmatrix} \longrightarrow u_k = \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix}$$

So:

$$u_0 = \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$u_1 = Au_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_1 + F_0 \\ F_0 \end{bmatrix}$$

Wait... that doesnt correspond to the fibonacci sequence, the second entry should have been F_{k+1}, therefore make sure to adjust A.

$F_{k+1} + F_k = F_{k+2}$

$F_{k+1} + 0 = F_{k+1}$

Perhaps the artifical equation should have its variable placed to the left-most pivot. Just make sure to double check matrix A fits the logic that you expect.

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

$$u_1 = Au_0 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} F_1 + F_0 \\ F_1 \end{bmatrix}$$

Now it make sense. And from here:

$$u_k = A^k u_0$$

We can look for the eigenvalues and eigenvectors of A:

$$|\, A - \lambda I \,| = \begin{vmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 - \lambda - 1 = 0$$

$$\lambda_1 = \frac{1}{2}(1+\sqrt{5}) \approx 1.618 \;\; \lambda_2 = \frac{1}{2}(1-\sqrt{5}) \approx -0.618$$

The eigenvalue controls the growth of the fibonacci sequence, namely the linear combination of them and the eigenvectors. Since the negative one is absoluteively (in the mathematical | | sense) smaller than 1, it will approximate 0 as k goes to infinity.

$$F_{100} = c_1 \lambda_1^{100} + c_2 \lambda_2^{100}$$

$$F_{100} \approx c_1 \left(\tfrac{1+\sqrt{5}}{2} \right)^{100} + 0$$

The eigenvectors can be solved using the techniques seen before.

Prey-predator system

Let vector x_k denote the population of both owls and rats after k months such that

$$x_k = \begin{bmatrix} O_k \\ R_k \end{bmatrix}$$

Then let the system below denote the population dynamics overtime:

$$\begin{cases} O_{k+1} = 0.5 O_k \; + (0.4) R_k \\ R_{k+1} = -(0.104) O_k + 1.1 R_k \end{cases}$$

- The first equation indicates that:
  - with no rats, only half the owls will survive each month
  - If there are plenty rats, the owl population will increase
- The second equation indicates that:
  - with no owls, the rats would increase by 10% each month
  - If there are plenty owls, the rat population will decrease
- The direction of greatest repulstion is the eigenvector that has eigenvalue greater than 1 (in absolute terms).

- The direction of greatest attraction is the eigenvector that has eigenvalue less than 1 (in absolute terms).

We can see from this system that there should be a sweetspot overtime in which there will be a balanced amount of owls and rats keeping both populations stable.

- That sweet spot is $\lim x_{k k\to\infty} \approx x^{10^{a\ lot}} = A^{10^{a\ lot}} x_0$
- A can be made from the system definition:

$$A = \begin{bmatrix} 0.5 & 0.4 \\ -0.104 & 1.1 \end{bmatrix}$$

Matlab returns the eigenvalues 0.58 and 1.02, and the following eigenvector matrix V:

```
>> A=[.5 .4; -.104 1.1]

A =

    0.5000    0.4000
   -0.1040    1.1000

>> [V,D] = eig(A)

V =

   -0.9806   -0.6097
   -0.1961   -0.7926


D =

    0.5800         0
         0    1.0200

>>
```

Matlab has rounded the the operations, but in fact -0.9806/-0.1961 = 5/1 and -0.6097/-0.7926 = 10/13.

Therefore, the corresponding eigenvectors for $\lambda_1 = .58$ and $\lambda_1 = 1.02$ are

$$v_1 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \quad v_2 = \begin{bmatrix} 10 \\ 13 \end{bmatrix}$$

Instead of using $x_k = A^k x_0$ directly, since we don't know x_0 yet, we can **decompose** x_0 in eigenvectors:

Eigenvector decomposition

$$Let\ x_0 = \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix}.$$

Then we have that

$$x_0 = c_1 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_2 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_3 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix}$$

Since we defined $x_k = Ax_{k-1}$, then we have that

$$x_1 = A\left(c_1 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_2 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_3 \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix}\right) = c_1 A \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_2 A \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} + c_3 A \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} = c_1 Av_1 + c_2 Av_2 + c_3 Av_3$$

- The thing is that x can't just be decomposed into arbitrary vectors and scalars. x_0 must be strictly expressed as a linear combination of the eigenvectors scaled by some constants c. To find those c constants first we must find the eigenvectors and eigenvalues of A (matlab: `[V,D] = eig(A)`). If x_0 is given in the problem, then we can make another system of equations to solve for x_0, i.e.

$$\begin{cases} c_1 v_{11} + c_2 v_{21} = x_{01} \\ c_1 v_{12} + c_2 v_{22} = x_{02} \end{cases}$$

- Solving this system gives you the vector c, which can be then used in the formula below:

Also, Since v are eigenvectors of A, we have $Av = \lambda v$:

$$x_1 = c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + c_3 \lambda_3 v_3$$

$$x_k = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + c_3 \lambda_3^k v_3$$

$$x_0 = c_1 \lambda_1^0 v_1 + c_2 \lambda_2^0 v_2 + c_3 \lambda_3^0 v_3 = c_1 v_1 + c_2 v_2 + c_3 v_3$$

Therefore, if in our owl vs rat example, since x_0 is unkown, we can't solve for c. We can only express the solution given $\lambda_1 = .58$, $\lambda_2 = 1.02$ and

$$v_1 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \quad v_2 = \begin{bmatrix} 10 \\ 13 \end{bmatrix}$$

Then leave c as a constant $x_k = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2$

$$x_k = c_1 .58^k \begin{bmatrix} 5 \\ 1 \end{bmatrix} + c_2 1.02^k \begin{bmatrix} 10 \\ 13 \end{bmatrix}$$

From the fact that .58 is smaller than 1, as k grows to infinity, the rate of growth for both owls and rats will be solely defined by 1.02 and the ratio of owls to rats will be mantained at 10/13. Although c is not known, as we get closer to infinty that becomess irrelevant, and we've managed to solve one of the things we managed to learn some things about the system nevertheless.

Complex systems

Take matrix A which as complex eigenvalues:

$$A = PCP^{-1}$$

- Where matrix C is strictly composed of the complex eigenvalues of matrix A, such that a is the real part of the eigenvalues and $\pm b$ is the coeffcient multiplying i in the complex eigenvalue.

$$C = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

- Rember that P is composed of the real parts on the 1st column, and the negative imaginary parts on the second column.
- Trying to do $A^k = P\left(\begin{bmatrix} a & -b \\ b & a \end{bmatrix}\right)^k P^{-1}$ is hard, because C is not a diagonal matrix
- Instead use the property:

$$C^k = r^k \begin{bmatrix} \cos(k\phi) & -\sin(k\phi) \\ \sin(k\phi) & \cos(k\phi) \end{bmatrix}$$

- Remember that $\phi = \arctan(\frac{b}{a})$ and that $r = |\lambda| = \sqrt{a^2 + b^2}$
- Therefore $x_k = \left(PC^kP^{-1}\right) x_0$ or, if you know the previous transformation $x_k = Cx_{k-1}$
- This transformation overtime is a rotation that creates:

- Spiral towards the origin if r < 1
- Spiral away from the origin if r > 1
- Elliptic (or even circle sometimes) if r = 1.

Orthogonality

- Orthogonal means 90 degrees and is described by the symbol $\perp$, also known as perpendicular
- Not just 2 vectors can be orthogonal when the dot product is 0, but also subspaces themselves (seen as areas) can be orthogonal
  - Examples are rowspace vs nullspace and columnspace vs nullspace

Inner product

- Also known as dot product between two vectors, it's equivalent to the matrix-vector product of one of the vectors transposed with the other one so that the result is a 1x1 matrix, interpreted as a scalar:

$$\vec{a} \bullet \vec{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \vec{a}^T\vec{b} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = b_1a_1 + b_2a_2$$

This transformation expresses the length of the the distance from the origin to the point where one vector is orthogonally projected on to the other one, times the length of the later one. It happens that when the vectors are already perpendicular, the projection point is the origin, and thus we get 0*length of the vector where the projection lies. When the dot product is negative it means they point in the opposite direction, when positive in the same one.

- A dot product is only defined when the vectors have the same number of entries. Let u, v and we be vectors and c a scalar, a dot product can be expressed with a dot or with the transponse:
- $u \bullet v = u^Tv$
- $u^Tv = v^Tu$
- $(u+v)^Tw = u^Tw + v^Tw$
- $(cu)^Tv = c(u^Tv) = u^T(cv)$
- $u^Tu > 0$, uu=0 only for u=0
- The length (called norm) of v is expressed as the nonnegative scalar $||v||$ defined by:
- $$||v|| = \sqrt{v^Tv}$$
- Geometric interpretation of the dot proudct:

$$a \bullet b = ||a||||b|| \cos \alpha$$

- The distance between two vectors is the length of $||u-v||$, also expressed as $dist(u,v) = \sqrt{(u-v) \bullet (u-v)}$

Normal vectors

- Vectors whose length are 1 are called unit vectors
  - normalizing: scale a nonzero vector v by $\frac{1}{||v||}$ to obtain a unit vector

Orthogonal vectors

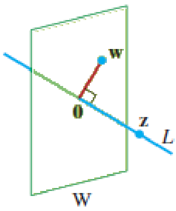
- Orthogonal vectors are vectors with a 0 innerproduct
  - Another test is based on pythagoras: u and v are orthogonal iff $||u+v||^2 = ||u||^2 + ||v||^2$
- The 0 vector is orthogonal with all the vectors, including itself

Orthonormal vectors

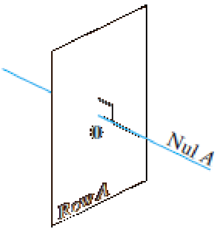
- Orthonormal vectors are orthogonal and normal (length 1) vectors

Orthogonal complements

- A vector z is orthogonal to a subspace W iff it is orthogonal to each vector in the subspace W set.
    - The set of all orthogonal vectors to W are called the **orthogonal complement** of W and is denoted as $W^\perp$



- If W is a plane, then $W^\perp$ is a line (all the vectors spanned by 1 basis vector perpendicular on W)
    - $W^\perp = L$ and $W = L^\perp$ (the complement of the complement cancels out)
        - under the premise that all the vectors must depart from the origin only
        - If 2 subspaces have an overlaping vector that is not the origin then they are not orthogonal as that vector cannot be orthogonal to itself (only 0 is orthogonal to itself)
- $\left(W^\perp\right)^\perp = W$
- For an n x m matrix:
    - dim Col A + dim $NulA$ = m, where dim stands for number of columns.
    -
- Rowspace is orthogonal to nullspace because of the defintion of the nullspace which takes all the x's such that Ax=0. This also means that each of the rows of A make a dot product with x and yield 0:
    - rowspace = rowspan:



- Dot product of each row with vector x (of the null space) yields zero:

$$Ax = 0 \Leftrightarrow \begin{bmatrix} row_1 \\ row_2 \\ \vdots \\ row_n \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- However, the subspace of the rows is not just the rows themselves, it's also all possible linear combinations of the rows. These are also orthogonal to the nullspace:

$$c_1 * row_1 = 0$$

$$c_2 * row_2 = 0$$

$$c_2 * row_2 + c_1 * row_1 = 0$$

- Since this statement is true for any matrix, it is also true for A^T. Therefore:
    - $(Row\ A)^\perp = NulA$
    - $(Row\ A^T)^\perp = NulA^T \iff (Col\ A)^\perp = NulA^T$

Orthogonal sets

A set of vectors { a_1, a_2, a_3, ..., a_n} in R^n is an **orthogonal set** if each possible pair combination of **distinct** vectors from the set is orthogonal, i.e.:

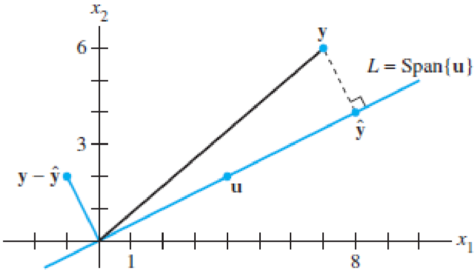
$$\{\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}\}$$

- If the orthogonal set does not contain the zero vector, then all the vectors are independent
- An **orthogonal basis** for a subspace W of R^n is a basis for W that is also an orthogonal set
- Orthogonal basis are **nice**:
  - any vector in R^n can be expressed as a linear combination of the columns of A
  - the weights of the columns for any vector y can be defined by
  - $$c_j = \frac{y \bullet a_j}{a_j \bullet a_j}$$
  - such that $y = c_1 a_1 + c_2 a_2 + \cdots + c_n a_n$
  - Therefore you dont need an augmented matrix to solve for the weights

Orthogonal projections

This is relevant when our matrix A has too many rows, and therefore some free variables.

The orthogonal projection $\hat{y}$ onto L (spanned by u) is a vector that lands on the point $(\hat{y}_1, \hat{y}_2)$. The line between the tip of y and the tip of $\hat{y}$ that connects these two vectors, in the minimum length possible, is is called "error" (e), as its length is the difference between them ($||y - \hat{y}||$), and the vector e comes from the arithmetic operation $y - \hat{y}$, which is ortogonal to the line L where projection $\hat{y}$ lies onto.



- We can see that $\hat{y}$ is a multiple of u, therefore: $\hat{y}$ = cu
- We can also see that $\hat{y} + e = y$
  - $e = y - \hat{y}$
- We also see that $L \perp e$, therefore: $0 = u^T e = u^T(y - \hat{y}) = u^T(y - cu) = 0$
  - $u^T y - cu^T u = 0$
  - $cu^T u = u^T y$
  - $c = \frac{u^T y}{u^T u}$

$$\hat{y} = \frac{u^T y}{u^T u} u$$

If I double y, I double $\hat{y}$ (it represents the same side of a twice as large equivalent triangle):

$$2\hat{y} = \frac{u^T 2y}{u^T u} u = 2\frac{u^T y}{u^T u} u$$

If I double v, $\hat{y}$ remains constant (it's projected onto the same line from the same spot):

$$p = \frac{2u^T y}{2u^T 2y} 2u = \frac{u^T y}{u^T u} u$$

This equation is formally known as the orthogonal projection of vector y onto a line L spanned by u, who both go through the origin:

$$\hat{y} = proj_L\, y = \frac{y \bullet u}{u \bullet u} u$$

Orthonormal sets

It is like an orthogonal set, but in addition the length of the vectors is 1. The clearest examples are a subset of vectors e (columns of I matrix) for R^n.

- An m x n matrix U has orthonormal columns iff $U^TU = I$
- U matrix with orthonormal sets of columns (this alone does not qualify for a orthogonal matrix), are even **nicer**, as they also have that:
  - The length of a transformation remains the same: $\|Ux\| = \|x\|$
  - Dot product between vectors is preserved: $(Ux) \bullet (Uy) = x \bullet y$
  - $(Ux) \bullet (Uy) = 0 \iff x \bullet y = 0$
  - Since the length of the vectors is 1, any vector y can be expressed as $y = (y \bullet a_1)a_1 + (y \bullet a_1)a_2 \ldots$

Orthogonal matrix

- Orthogonal matrices are actually **orthonormal column** square matrices
- An orthogonal matrix is a **square, invertible** matrix Q with **orthonormal columns**, such that:
  - $Q^{-1} = Q^T$
  - $Q^TQ = I$
- Orthogonal matrixes have orthonormal rows too
- Transformations with an orthonormal matrix makes the projection onto its column space easy since Q is **squared** ($QQ^T = Q^TQ = I$):
  - $P = Q(Q^TQ)^{-1}Q^T = QQ^T = I$
  - Doesnt work for non square matrices as $2x3 \cdot 3x2 = 3x3 \; vs \; 3x2 \cdot 3x2 = 2x2$
- Their determinant is either 1 or -1
- To prove that a matrix is orthogonal just prove that inverse is equal to the transponse or that $Q^TQ = I$

Orthogonal Projections in higher dimensions

- In higher dimensions instead of having a single vector u span a subspace of R^2, namely the line L, we can have an orthogonal basis set composed of { $u_1, u_2, \ldots, u_n$} that spans a subspace W.
- The orthogonal projection of y onto W is $\hat{y}$, which is the closest point to y in that lies in W:
  - $\hat{y}$ makes the smallest error (difference of y vs a vector in W): $\|y - \hat{y}\| < \|y - v$, where v represents any other vector in W distinct from $\hat{y}$.
  - $\hat{y}$ is formally regarded as the best approximation to y by elements of W.
  - It doesnt matter which orthogonal basis is used for W (in 2D we just scaled u by 2), as the projection point will lie on the same spot in higher dimensions too. However, regardless of the basis for W, the subspace must remain the same.

This projection of y onto line L spanned by u can be translated into a matrix, the **projection matrix**: $\hat{y} = Py$, where P stands for the projection matrix

$$P = \frac{u}{u^Tu}u^T$$

- We are left with a transposed vector, which is equivalent to a 1xn matrix
- Remember that $\frac{1}{A} = A^{-1}$, therefore:

$$P = U(U^TU)^{-1}U^T$$

- Applying the projection a second time will land you on the same spot you already are from the first projection.
- $P^2 = P$

For higher dimensions, we have more vectors in the orthogonal basis. Therefore P looks like:

$$P = \frac{u_1}{u_1^Tu_1}u_1^T + \frac{u_2}{u_2^Tu_2}u_2^T + \cdots + \frac{u_n}{u_n^Tu_n}u_n^T$$

Therefore:

$$\hat{y} = Py = (\frac{u_1}{u_1^T u_1} u_1^T + \frac{u_2}{u_2^T u_2} u_2^T + \cdots + \frac{u_n}{u_n^T u_n} u_n^T) y$$

```
projection=((v1/(transpose(v1)*v1))*transpose(v1)+(etcera))*y
```

Orthonomal basis projections

If $P = \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix}$, that is, the columns of P are an orthonormal basis. Then the lengths of the columns are all 1 and the projection of y onto subspace W is simplified to:

$$proj_W y = UU^T y$$

If P is a squared matrix, it happens that all the u columns span the entire space R^n already. Therefore any vector that you are going to "project" onto the "subspace" of R^n is already in R^n, so the vector is not transformed at all and $P = I$.

- This matches the fact that for orthogonal matrices (squared matrices with orthonormal basis): $QQ^T = Q^T Q = I$

Gram-Schmidt process

- Orthogonal projections onto a subspace W can only be made with orthogonal basis vectors. If the given subspace is not defined with orthogonal vectors, then you cant use the known techniques to project y onto W (z or e are not orthogonal to W).
- The gram-schmidt process is to find an equivalent basis vectors that are orthogonal:

Given a basis $\{x_1, \dots, x_p\}$ for a nonzero subspace W of R^n:

$$v_1 = x_1$$

$$v_2 = x_2 - \frac{x_2 \bullet v_1}{v_1 \bullet v_1} v_1$$

$$v_3 = x_3 - \frac{x_3 \bullet v_1}{v_1 \bullet v_1} v_1 - \frac{x_3 \bullet v_2}{v_2 \bullet v_2} v_2$$

Then $\{v_1, \dots, v_p\}$ is an orthogonal basis for W.

This formula formally means:

$$v_1 = x_1$$

$$v_1 = x_2 - projX_{1_{Span\{x_1\}}}$$

$$v_n = x_n - projX_{n_{Span\{x_1, \dots, x_n\}}}$$

- Span $\{x_1, \dots, x_p\}$ = Span $\{v_1, \dots, v_p\}$.

Orthogonal to orthonormal

This is an easy transformation. (Often after doing grand-schmidt), just scale the vectors to be normal vectors (length 1):

- Scale a nonzero vector v by $\frac{1}{\|v\|}$ to obtain a normalized vector.

Least squares

- If you can't solve Ax=b because the system has no solution. You may as well just go for the next best thing, which is to project b onto the columnspace of A, and then solve for $Ax = \hat{b}$ which does have a solution. To do so A must have orthogonal columns.

- Formally we would want to change the name of x for $\hat{x}$, which stands for the approximated x that solves for the the projected b, called $\hat{b}$, such that $\hat{b} = projb_{ColSpaceA}$. However, we are not manually projecting x anywhere, it's just to highlight that it's a least squares solution associated with the projected b.
- This creates the inequality equation below for a matrix A in m x n with a b in R^m, but whose columns can only Span R^n:

$$||b - A\hat{x}|| \leq ||b - Ax||$$

"for all x in R^n". Which means:

$$||b - projX| \leq ||b - Ax||$$

$$||e_{\hat{x}}| \leq ||e_{any\, x}||$$

- This is true because the error associated with the projected b solution: $e_{\hat{b}} \perp ColA$, whereas the other choices of x do not make a perpendicular e, which makes $||e_{\hat{b}}||$ be the shortest length, called the least-squares because this length is calculated by the square root of the sums of the squared entries. Some processes skip the square root step. This is almost the same as the "best approximation theorem".
  - The least squares solution is in $R^{number\ of\ columns}$
  - The least squares solutions is nonempty
  - $||b - A\hat{x}||$ is caled the least-squares error.

QR Factorization

- Any m x n matrix with linearly independent columns can be factored as A = QR, where Q is an m x n orthonormal basis (apply grand-schmidt of A for that) and R is the n x n upper triangular invertible matrix with positive entries on its diagonal that follows from:
- $A = QR$
- $Q^T A = Q^T QR = IR$
- $Q^T A = R$
- The solution for least squares: $\hat{x} = R^{-1} Q^T b$
- It is faster to solve for: $Rx = Q^T b$

Least squares when A is already orthogonal

The shortcut $Rx = Q^T b$ is not needed when A is already orthogonal. You can calculate the projection $\hat{b}$ like in Orthogonal Projections in higher dimensions and solve for $Ax = \hat{b}$.

Least squares shortcut for all matrices with A linearly independent and (A^T)A invertible

These statementes are all logically equivalent:

- The columns of A are linearly independent
- The matrix $A^T A$ is invertible
- Ax=b has a unique least squares solution for each b, precisely:

$$\hat{x} = (A^T A)^{-1} A^T b$$

- All the solutions for $A\hat{x} = \hat{b}$ (formally known as the least squares solutions for Ax=b). Are the exact same solutions as for $A^T Ax = A^T b$, known as the "normal equation". Which is easier than to have to project b and then solve x for $Ax = \hat{b}$.
  - Since $e_{\hat{b}} \perp ColA \Leftrightarrow (b - \hat{b}) \perp ColA \Leftrightarrow (b - A\hat{x}) \perp ColA$
    - $a_1 \bullet (b - A\hat{x}) = 0$
    - $a_2 \bullet (b - A\hat{x}) = 0$
    - ...
    - $a_n \bullet (b - A\hat{x}) = 0$
    - $A^T (b - A\hat{x}) = 0$

- $A^Tb - A^TA\hat{x} = 0$
- $A^Tb = A^TA\hat{x}$
- $\hat{x} = (A^TA)^{-1}A^Tb$

Solution when (A^T)A is not invertible: Use the normal equation

- Then the columns of A are not linearly independent
- There are more than one least square solutions
- You cant do QR Factorization nor the shortcut for linearly independent matrices. Solutions can be found with the "normal equation": $A^TAx = A^Tb$

Linear models

A linear model is used to "model" (pretend) a relationship between data sets by means of a formula that predicts the value of one variable as a function of other variable(s). A linear model is expresed as a system of linear equations defined by the data. In the real world, data collection might come with some error/noise and/or the variables themselves are not linearly related per se. However, to simplify and estimate things, it is possible to find a linear equation with the smallest error (as in distance from linear model to actual data points), for which we use Least squares. **When only y is presumed to have error the residual is no longer the perpendicular line from the observed value to the predicted value (from b to projected b), but just the difference in the y-axis, that is, the residual is** $y_n - (\beta_0 + \beta_1 x_1)$

- Notation differs between Linear algebra and Statistics:

Linear algebra	Statistics
$Ax = b$	$X\beta = y$
Matrix A	Design matrix X
input vector x	parameter vector (regression coeffiecients) β
output vector b	observation vector y

Least-Squares Lines

y=ax + b is expressed ass $y = \beta_0 + \beta_1 x$.

Fitting line to a 2D model data

Data for which we have $(x_1, y_1)\dots(x_n, y_n)$ data points and a model $y = \beta + \beta_1 x$ (also known as the **regression line of y on x**, because errors in data are assumed to be only on y… that is, it is clear that x is the input, independent variable and y the dependent variable, we can distinguish between the **predicted y-value** given by the regression line: $(x_j, \beta_0 + \beta_1 x_j)$. and the actual observed value of y_j. The difference between the observed value (in LA: y) and the predicted value (in LA $\hat{y}$) is called the residual (in LA the error, however the error in LA is exclusively the perpendicular one, whereas in this particular regression case is just the difference of the y-axis values. Although the interpetation for error may vary, the approach to approximate for beta and y is the same).

Measuring how close is the line to the data

- Easiest choice is to add the squares of the residuals as it produces an absolute non-negative length.
- The least-squares is the line $y = \beta_0 + \beta_1 x$ that minimizes the sum of the squares of the residuals.

Translating data points into a system

- If data points were on a straight line, the parameters $\beta_0 + \beta_1$ would satisfy:

Predicted y-value	Observed y-value
$\beta_0 + \beta_1 x_1$	y_1
$\beta_0 + \beta_1 x_2$	y_2
$\vdots$	$\vdots$
$\beta_0 + \beta_1 x_n$	y_n

- Such that $X\beta = y$, where:

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

If the data points don’t lie on a line, then there is no solution for β such that the entries of β make the matrix vector product output y. However, we can apply least squares to generate a projection of y onto the column space of X, and find the beta for that equation.

The general linear model

- To make a models that use more variables we an use the same equation by just expanding the columns of X and the entries of β accordingly. Statisticans add an additional vector, the residual vector ϵ, such that:
- $y = X\beta + \epsilon$
- It is still a linear model, this line just travels in higher dimensions.
- X and y are determined by the data set. β and the residual vector are found via least-squares.
- The least square solution β is a solution of the normal equations:

$$X^T X\hat{\beta} = X^T \hat{y}$$

Least-squares fitting for other curves

- When the data does not seem to fit into a line we can still make a model
- the unkowns do not necessarily need to be a “raw” variable (such as $x_1, x_2 \ldots$)
- The unkowns can be functions (such as $f_1(x), f_2(x) \ldots$)
- This is a “fit” to a curve for instance.
- Usually the shape of the data distribution reveals a likely model to use to try to minimize the resiudals of, such as the formula below for parabolas:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2$$

- Then we can turn it into a system of equations based on each data point:

$$y_1 = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \epsilon_1$$

$$y_2 = \beta_0 + \beta_1 x_2 + \beta_2 x_2^2 + \epsilon_2$$

$$\vdots$$

$$y_n = \beta_0 + \beta_1 x_n + \beta_2 x_n^2 + \epsilon_n$$

- Such that $y = X\beta + \epsilon$ and:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \\ 1 & x_n & x_n^2 \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \end{bmatrix}$$

- Looking at the graph and (arbitrarily) deciding to model it based on a quadratic formula is adecuate as long as you are happy with the size of the residual. Sometimes throwing more columns into X might lead to smaller residuals.

Multiple Regression

These are models that have multiple variables as input, which then make the model a 3D or more model. The model can be made in the same fashion, just that the unkowns are functions that take 2 or more inputs, such that:

$$y = \beta_0 f_0(u, v) + \beta_1 f_1(u, v) + \dots$$

or also as a simpler fashion such as:

$$y = \beta_0 + \beta_1 u + \beta_2 v \dots$$

The least squares process to fit the data is called **trend surface**. An example would be a model that fits data points for altitude, latitud and longitude.

Symmetric matrices

- The matrix is symetric with respect to the main diagonal
- Therefore $A = A^T$
  - Such matrix must be square
  - Main diagonal is arbitrary, the other entries occur in pairs (on opposite sides of the main diagonal)
- All eigenvalues are real
- For each eigenvalue: the geometric multiplicity = algebraic multiplicity
- Eigenvectors of a symmetric matrix that correspond to different eigenvalues are orthogonal
  - 2 distinct eigenvectors can have the same eigenvalue scalar, yet these 2 eigenvectors are still different and not necessarily orthogonal, but for this eigenspace there exists an orthonormal basis. Use Grand-Schmidt.
- The inverse of an orthogonal matrix (such as the eigenvector matrix of a symmetric matrix) is the transpose
  - Which makes symetric matrices very easy to diagonalize
  - Usually we express $A = S\Lambda S^{-1}$, but now we can normalize S, such that $A = Q\Lambda Q^{-1} = Q\Lambda Q^T$
  - A is a symmetric matrix iff A is orthogonally diagonalizable
- The number of positive eigenvalues is the same as the number of positive pivots

Sepctral theorem

- The set of eigenvalues of A is sometimes called the spectrum of A
- A has n real eigenvalues if we count the multiplicites
- The geometric multiplicity of an eigenvalue is equal to its algebraic multiplicity
  - That is, if the algebraic multiplicity is 2, then the eigenvalue Spans 2D
- The egienspaces are mutually orthogonal (because the eigenvectors are orthogonal)
- A is orthogonally diagonalizable

Sepctral decomposition

$$A = QDQ^T = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} q_1^T \\ \vdots \\ q_n^T \end{bmatrix}$$

$$A = \lambda_1 q_1 q_1^T + \lambda_2 q_2 q_2^T + \cdots + \lambda_n q_n q_n^T$$

- Each term has rank 1 and a multiple of u_n
- The vector $(u_j u_j^T)x$ is the orthogonal projection of x onto the subspace spanned by u_j.

Algorithm for orthogonal diagonalization

1. Compute the eigenvalues of A
2. Construct a basis of the corresponding eigenspace
3. Make the basis orthonormal
    1. Use Gram-Schmidt to rothogonalize if necessary
    2. USe scaling to normalize basis vectors
4. Construct the matrix S using the basis vectors
5. Construct matrix D using the eigenvalues on the main diagonal