LINEAR ALGEBRA (MATRIX) REVIEW

A.1: Terminology and notation

- Kalman filters require that the system whose state is being estimated be represented by a model in “state space” form.
- This form, and hence the Kalman filter method, relies *heavily* on matrix/vector operations.
- This set of notes reviews the mechanics of matrix manipulation. An attempt is also made to aid intuition.
- A matrix is a rectangular array of scalars written between brackets.

EXAMPLE:

$$A = \begin{bmatrix} 0.1 & 1.2 & -2.4 & 0.4 \\ 0.5 & -0.2 & 1.3 & -2.5 \\ -0.2 & 1.1 & 9.5 & -1.8 \end{bmatrix}.$$

- An important attribute of a matrix is its size or dimension:
- Always measured in number of rows × number of columns. Above: 3×4.
- The entries or coefficients are the values in the array.
- The i, j entry is the value in the ith row and the jth column.
- The i, jth entry in matrix A is A_{ij} which is a number.
- The positive integers i and j are called the (row and column, respectively) indices.

- In MATLAB:

```
A = [0.1, 1.2, -2.4, 0.4; 0.5, -0.2, 1.3, -2.5; -0.2, 1.1, 9.5, -1.8];
```

EXAMPLE: $A_{13} = -2.4$, $A_{31} = -0.2$. The row index of the bottom row is 3, the column index of the first column is 1.

- In MATLAB:

```
result = A(1,3);
```

- A matrix with only one column (*i.e.*, size of $n \times 1$) is called a column vector, or just a vector.
- Sometimes, size is specified by calling it an n-*vector*.
- Entries are denoted with just one subscript (the other is "1") as in v_3.
- The entries are sometimes called the *components* of the vector.

EXAMPLE:

$$v = \begin{bmatrix} 3 \\ 0.5 \\ -1 \end{bmatrix}$$

is a 3-vector (or 3×1 matrix); its third component is $v_3 = -1$.

- Similarly, a matrix with a single row (size $1 \times n$) is a *row vector*.

EXAMPLE:

$$w = \begin{bmatrix} 8 & -1 & 0.1 \end{bmatrix}$$

is a row vector (or 1×3 matrix); its second component is $w_2 = -1$.

- Sometimes a 1×1 matrix is considered to be the same as a scalar, *i.e.*, a number.
- Two matrices are *equal* if they are the same size and all the corresponding entries (which are numbers) are equal.

Notational conventions

- Some authors try to use notation that helps the reader distinguish between matrices, vectors and scalars.
- For example,
  - Greek letters ($\alpha,\ \beta,\ \ldots$) might be used for numbers;
  - Lowercase letters ($a,\ x,\ y,\ \ldots$) might be used for vectors;
  - Uppercase letters ($A,\ B,\ \ldots$) for matrices.
- Other notational conventions include matrices given in bold font ($\mathbf{H}$), or vectors written with arrows above them ($\vec{a}$).
- But, there are about as many notational conventions as authors!
- Be prepared to figure out what things are (*i.e.*, scalars, vectors, matrices) despite the author's notational scheme (if any exists!).

Zero matrices

- The zero matrix (of size $m \times n$) has all entries equal to zero.
- Sometimes written as $0_{m\times n}$ where subscript denotes size.
- Often just written as 0, the same symbol used to denote the number 0.
- You need to figure out the size of the zero matrix *from the context.*
- Zero matrices of different sizes are different matrices, even though we use the same symbol (*i.e.*, 0).
- In programming, this is called overloading; we say that the symbol 0 is overloaded because it can mean different things depending on its context (*i.e.*, the equation it appears in).

- When a zero matrix is a (row or column) vector, we call it a zero (row or column) vector.

Identity matrices

- An identity matrix is another common matrix.
- It is always square, *i.e.*, has the same number of rows as columns.
- Its diagonal entries, *i.e.*, those with equal row and column indices, are all equal to 1.
- Its off-diagonal entries, *i.e.*, those with unequal row and column indices, are equal to 0.
- Identity matrices are denoted by the letter I. Sometimes a subscript denotes the size, as in I_3 or maybe $I_{2\times 2}$.
- Often, size must be determined from context (as for zero matrices).
- Formally, the identity matrix is defined by

$$I_{ij} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

EXAMPLES:

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

which are 2×2 and 4×4 identity matrices. Remember that both are denoted by the same symbol—I.

- In MATLAB:

```
I2 = eye(2);
```

A.2: Matrix operations: Transpose, addition, subtraction

- Matrices may be manipulated and combined in various operations to form other matrices.

Matrix transpose

- If A is an $m \times n$ matrix, its transpose, denoted A^T (or sometimes A'), is the $n \times m$ matrix given by

$$\left[A^T\right]_{ij} = A_{ji}.$$

- In other words, the rows and columns of A are transposed in A^T.

EXAMPLE:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 & 1 \\ 2 & 5 & 8 & 2 \\ 3 & 6 & 9 & 3 \end{bmatrix}.$$

- Note that transposition converts row vectors into column vectors, and vice versa.
- If we transpose a matrix twice, we get back the original matrix: $\left(A^T\right)^T = A$.
- If a matrix equals its own transpose, that matrix is said to be symmetric. Symmetric matrices will come up a lot in this course, and have special properties that we will find very useful.
- In MATLAB, there are two different transpose operators:

```
Atranspose1 = A'; % Conjugate (Hermetian) transpose
Atranspose2 = A.'; % True transpose
```

Matrix addition and subtraction

- Two matrices *of the same size* can be added together to form another matrix (of the same size) by adding the corresponding entries.
- Matrix addition is denoted by the symbol $+$. (Thus the symbol $+$ is overloaded to mean scalar addition when scalars appear on its left- and right-hand side, and matrix addition when matrices appear on its left- and right-hand sides.)

EXAMPLE:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}.$$

- Note that (row or column) vectors of the same size can be added, but you cannot add together a row vector and a column vector (except if they are both scalars!).
- Matrix subtraction is similar:
- In MATLAB:

```
result = [1 2; 3 4] + [5 6; 7 8];
```

EXAMPLE:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - I = \begin{bmatrix} 0 & 2 \\ 3 & 3 \end{bmatrix}.$$

- Note that this gives an example where we have to figure out what size the identity matrix is. Since you can only add (or subtract) matrices of the same size, we conclude that I must refer to a 2×2 identity matrix.
- In MATLAB:

```
result = [1 2; 3 4] - eye(2);
```

- Matrix addition is commutative; *i.e.*, if A and B are matrices of the same size, then $A + B = B + A$.
- It is also associative; *i.e.*, $(A + B) + C = A + (B + C)$, so we write both as $A + B + C$.
- We always have $A + 0 = 0 + A = A$; *i.e.*, adding the zero matrix has no effect.

Scalar multiplication

- If we multiply a matrix by a scalar, the resulting matrix has every entry multiplied by the scalar.
- Usually denoted by juxtaposition, with the scalar on the left, as in

$$(-2)\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} -2 & -8 \\ -4 & -10 \\ -6 & -12 \end{bmatrix}.$$

- In MATLAB:

```
result = (-2) * [1 4; 2 5; 3 6];
```

- Sometimes you see scalar multiplication with the scalar on the right, or even scalar division with the scalar shown in the denominator (which just means scalar multiplication by one over the scalar), as in

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \cdot 2 = \begin{bmatrix} 2 & 8 \\ 4 & 10 \\ 6 & 12 \end{bmatrix}, \qquad \frac{\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}}{2} = \begin{bmatrix} 0.5 & 1.5 & 2.5 \\ 1 & 2 & 3 \end{bmatrix},$$

but these look ugly.

- Scalar multiplication obeys several laws you can determine for yourself. *e.g.*, if A is any matrix and $\alpha,\ \beta$ are any scalars, then

$$(\alpha+\beta)A=\alpha A+\beta A.$$

- It is useful to identify the symbols above. The $+$ sign on the left is the addition of scalars. The $+$ sign on the right denotes matrix addition.
- Another simple property is $(\alpha\beta)A=(\alpha)(\beta A)$, where α and β are scalars and A is a matrix. On the left side we have scalar-scalar multiplication $(\alpha\beta)$ and scalar-matrix multiplication; on the right side we see two cases of scalar-matrix multiplication.
- Note that $0\cdot A=0$ (where the left-hand zero is the scalar zero, and the right-hand zero is a matrix zero of the same size as A).

A.3: Matrix operations: Multiplication, powers, inverse

Matrix multiplication

- It is also possible to multiply two matrices using matrix multiplication.
- You can multiply two matrices A and B provided that their dimensions are compatible*,* which means that the number of columns of A equals the number of rows of B.

EXAMPLE:

$$A_{m\times p}B_{p\times n} = C_{m\times n}.$$

- The product is defined by

$$C_{ij} = \sum_{k=1}^{p} A_{ik}B_{kj} = A_{i1}B_{1j} + \cdots + A_{ip}B_{pj}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

- This looks complicated, but is not too difficult.

$$\begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & & \vdots \\ A_{i1} & \cdots & A_{ip} \\ \vdots & & \vdots \\ A_{m1} & \cdots & A_{mp} \end{bmatrix} \begin{bmatrix} B_{11} & \cdots & B_{1j} & \cdots & B_{1n} \\ \vdots & & \vdots & & \vdots \\ B_{p1} & \cdots & B_{pj} & \cdots & B_{pn} \end{bmatrix} = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & & \vdots \\ & C_{ij} & \\ \vdots & & \vdots \\ C_{m1} & \cdots & C_{mn} \end{bmatrix}$$

- To find the i, jth entry of the product $C = AB$, you need to know the ith row of A and the jth column of B.
- The summation above can be interpreted as “moving left-to-right along the row i of A while moving top-to-bottom down column j of B. As you go, keep a running sum of the product of entries: one from A and one from B.”

- Now we can explain why I is called the identity matrix: If A is any $m \times n$ matrix, then $AI = A$ and $IA = A$, *i.e.*, when you multiply a matrix by an identity matrix, it has no effect. (The identity matrices in $AI = A$ and $IA = A$ have different sizes—what are they?)
- One *VERY* important fact is that matrix multiplication is not (in general) commutative. We *DON'T* have $AB = BA$. In fact BA may not even make sense (due to dimensions) and even if it does make sense, it may have different dimension than AB so that equality in $AB = BA$ is meaningless.

EXAMPLE: If A is 2×3 and B is 3×4 then AB makes sense, and is 2×4. BA does not make sense.

EXAMPLE: Even if both make sense (as in when both are square, for example) $AB \neq BA$ in general

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}, \qquad \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 23 & 34 \\ 31 & 46 \end{bmatrix}.$$

- In MATLAB:

```
result = [1 2; 3 4] * [5 6; 7 8];
```

- Matrix multiplication is associative; *i.e.*, $(AB)C = A(BC)$. Therefore, we write a product as ABC.
- Matrix multiplication is also associative with scalar multiplication; *i.e.*; $\alpha(AB) = (\alpha A)B$.
- Matrix multiplication distributes across matrix addition: $A(B + C) = AB + AC$, and $(A + B)C = AC + BC$.

Matrix–vector product

- A very important type of matrix multiplication: matrix-vector product.

EXAMPLE:

$$y = Ax,$$

where A is an $m \times n$ matrix, x is an n-vector and y is an m-vector.

- We can think of matrix-vector multiplication (with an $m \times n$ matrix) as a function that transforms n-vectors into m-vectors. The formula is:

$$y_i = A_{i1}x_1 + \cdots + A_{in}x_n, \quad i = 1, \ldots, m.$$

Inner product

- Another special case is the product of a row vector with a column vector with the same number of elements.
- Then vw makes sense, and has size 1×1 (*i.e.*, a scalar).
- $vw = v_1w_1 + \cdots + v_nw_n$. This often occurs in the form x^Ty where x and y are both column n-vectors. In this case the product is called the *inner product* or *dot product* of the vectors x and y. Other notation is: $\langle x, y\rangle$ or $x \cdot y$.

Matrix powers

- When a matrix A is square, then it makes sense to multiply A by itself; *i.e.*, to form $A \cdot A$. We call this A^2. Similarly, k copies multiplied together are A^k.
- Non-integer powers, such as $A^{\frac{1}{2}} = \sqrt{A}$, are pretty tricky.
  - They might not make sense, or be ambiguous, unless certain conditions on A hold.

  - We will need to be able to compute $\sqrt{A}$ later, and will discuss it more at that point in time.

- By convention, we set $A^0 = I$ (usually only when A is invertible).

Matrix inverse

- If A is square, and there is a matrix F such that $FA = I$, then we say that A is *invertible* or *nonsingular.* We call F the *inverse* of A, and denote it A^{-1}. Then, $A^{-k} = (A^{-1})^k$.

- It is important to note that not all square matrices are invertible. For example, the zero matrix never has an inverse. A less-obvious example is to show that

$$\begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix}$$

  does not have an inverse.

- As an example of a matrix inverse, we have

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

  (you should check this!)

- In MATLAB:

```
result = inv([1 1; 1 2]);
```

- It is very useful to know the general formula for a 2×2 matrix inverse.

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

  provided $ad - bc \neq 0$. (If $ad - bc = 0$, the matrix is not invertible.)

- When a matrix is invertible, $\left(A^{-1}\right)^{-1} = A$.

Useful identities

- Here are a few useful identities. This list is not complete!

1. Transpose of product: $(AB)^T = B^T A^T$.
2. Transpose of sum: $(A + B)^T = A^T + B^T$.
3. Inverse of product: $(AB)^{-1} = B^{-1}A^{-1}$ provided A and B are square and invertible.
4. Products of powers: $A^k A^l = A^{k+l}$ (for $k, l \geq 1$ in general, and for all k, l if A is invertible).

Block matrices and submatrices

- Sometimes it is convenient to form matrices whose entries are themselves matrices.

$$\left[\begin{array}{c|c|c} A & B & C \end{array}\right]$$

$$\left[\begin{array}{c|c} F & I \\ \hline 0 & G \end{array}\right].$$

where $A,\ B,\ C,\ F$ and G are matrices (as are 0 and I). Such matrices are called *block matrices.*

- Block matrices need to have the right dimensions to fit together.

EXAMPLE:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

$$\left[\begin{array}{c:c} A & B \\ \hdashline C & D \end{array}\right] = \left[\begin{array}{cc:c} 1 & 2 & 3 \\ 0 & 2 & 1 \\ \hdashline 1 & 0 & 0 \end{array}\right].$$

- Block matrices may be added and multiplied as if the entries were numbers, provided the corresponding entries have the right size and you are careful about the order of multiplication.

$$\left[\begin{array}{c:c} A & B \\ \hdashline C & D \end{array}\right] \left[\begin{array}{c} X \\ \hdashline Y \end{array}\right] = \left[\begin{array}{c} AX + BY \\ \hdashline CX + DY \end{array}\right],$$

provided the products AX, BY, CX and DY make sense.

- In MATLAB:

```
bigMatrix = [A, B; C, D];
```

A.4: Linear equations and matrices

Linear functions

- Suppose function f takes n-vectors as input and returns m-vectors.
- We say that f is linear iff

  SCALING: For any n-vector x and scalar α, $f(\alpha x) = \alpha f(x)$.

  SUPERPOSITION: For any n-vectors x and y, $f(x+y) = f(x) + f(y)$.
- Such a function may always be represented as a matrix-vector multiplication $f(x) = Ax$.
- Conversely, all functions represented by $f(x) = Ax$ are linear.
- We can also write the function in explicit form, where $f(x) = y$ as

$$y_i = \sum_{j=1}^{n} A_{ij} x_j = A_{i1} x_1 + \cdots + A_{in} x_n, \qquad i = 1, \ldots, m.$$

  This gives a simple interpretation of A_{ij}: it gives the coefficient by which y_i depends on x_j.

Linear equations

- Any set of m linear equations in (scalar) variables $x_1, \ldots\ x_n$ can be represented by $Ax = b$, where x is a vector made from the variables, A is a $m \times n$ matrix and b is a m-vector.

EXAMPLE:

$$1 + x_2 - x_3 = -2x_1, \qquad x_3 = x_2 - 2.$$

- Rewrite the equations with the variables lined up in columns, and the constants on the right-hand side.

$$\begin{aligned} 2x_1 \; &+x_2 \; -x_3 = -1 \\ 0x_1 \; &-x_2 \; +x_3 = -2. \end{aligned}$$

- Now it is easy to rewrite the equations as a single matrix equation

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

so we have two equations in three variables as $Ax = b$ where

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ -2 \end{bmatrix}.$$

Solving linear equations

- Suppose we have n linear equations in n variables $x_1, \ldots, x_n$, written in the compact matrix notation $Ax = b$.
- A is a $n \times n$ matrix; b is an n-vector. Suppose that A^{-1} exists. Multiply both sides of $Ax = b$ by A^{-1}.

$$\begin{aligned} A^{-1}(Ax) &= A^{-1}b \\ Ix &= A^{-1}b \\ x &= A^{-1}b. \end{aligned}$$

- We have solved the simultaneous equations.
- We can't *always* solve n simultaneous equations for n variables. One or more of the equations may be redundant (*i.e.*, may be obtained from the others), or the equations may be inconsistent (*i.e.*, $x_1 = 1$, and $x_1 = 2$).

- When these pathologies occur, A is singular (non-invertible). Conversely, when A is non-invertible, the equations are either redundant or inconsistent.

- From a practical point of view, either you don't have enough equations or you have the wrong ones. Otherwise, A^{-1} exists, and you can solve $x = A^{-1}b$.

Solving linear equations in practice

- When we solve linear equations by computer, we don't use $x = A^{-1}b$, although it would work. Practical methods compute $x = A^{-1}b$ directly.

- A may be large, sparse, or poorly conditioned. There exist efficient methods to handle each case.

- In MATLAB,

```
x=A\b;
```

The determinant function

- Consider the set of equations

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

or

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

- Multiply the first equation by a_{22} and the second equation by $-a_{12}$. Add the resulting two equations:

$$\begin{array}{r}
a_{11}a_{22}x_1 + a_{12}a_{22}x_2 = a_{22}b_1 \\
-a_{12}a_{21}x_1 - a_{12}a_{22}x_2 = -a_{12}b_2 \\
\hline
(\underbrace{a_{11}a_{22} - a_{12}a_{21}}_{\det(A)})x_1 = a_{22}b_1 - a_{12}b_2
\end{array}$$

so we can solve for x_1.

- Multiply first equation by $-a_{21}$ and the second by a_{11}. Add the resulting two equations:

$$\begin{array}{r}
-a_{11}a_{21}x_1 - a_{12}a_{21}x_2 = -a_{21}b_1 \\
a_{11}a_{21}x_1 + a_{11}a_{22}x_2 = a_{11}b_2 \\
\hline
(\underbrace{a_{11}a_{22} - a_{12}a_{21}}_{\det(A)})x_2 = a_{11}b_2 - a_{21}b_1
\end{array}$$

so we can solve for x_2.

- Determinants come up naturally when solving systems of equations.

$$2 \times 2: \qquad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$3 \times 3: \qquad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{array}{c} a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} \\ +a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} \\ -a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}, \end{array}$$

and so forth.

- In MATLAB:

```
result = det(A);
```

- $A^{-1} = \dfrac{\text{adj}(A)}{\det(A)}$, $\det(AB) = \det(A)\det(B)$ when both are square, and $\det(A^T) = \det(A)$.

A.5: Interpreting $y = Ax$

- Consider the system of linear equations

$$\begin{aligned} y_1 &= A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ y_2 &= A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ &\vdots \\ y_m &= A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{aligned}$$

which can be written as $y = Ax$, where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \qquad A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

- Some interpretations of $y = Ax$:
- y is measurement or observation; x is unknown to be determined
- x is “input” or “action”; y is “output” or “result”.
- $y = Ax$ defines a function that maps $x \in \mathbb{R}^n$ into $y \in \mathbb{R}^m$.
- A_{ij} is the gain factor from the jth input (x_j) to the ith output (y_i).
- Thus,
  - ith row of A concerns ith output.
  - jth column of A concerns jth input.
  - $A_{27} = 0$ means 2nd output (y_2) doesn’t depend on 7th input (x_7).
  - $|A_{31}| \gg |A_{3j}|$ for $j \neq 1$ means y_3 mostly depends on x_1.
  - $|A_{52}| \gg |A_{i2}|$ for $i \neq 5$ means x_2 mostly affects y_5.

Interpreting $y = Ax$ via the columns of A: a_i

- Write A in terms of its columns

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix},$$

where $a_j \in \mathbb{R}^m$.

- Then, $y = Ax$ can be written as

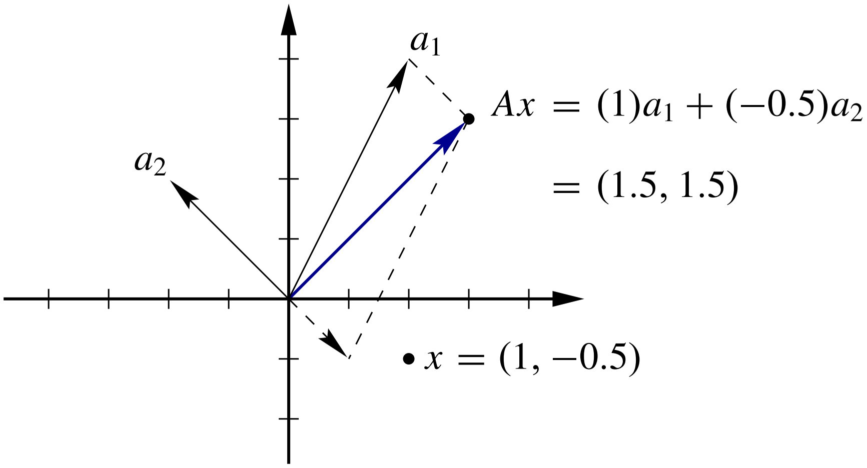
$$y = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n$$

(note: x_j's are scalars, a_j's are m-vectors)

- y is a linear combination or mixture of the columns of A.
- Coefficients of x give coefficients of mixture.

EXAMPLE:

$$A = \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}, \qquad x = \begin{bmatrix} 1.0 \\ -0.5 \end{bmatrix}, \qquad y = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}.$$



Interpreting $y = Ax$ via the eigenvectors/eigenvalues of A

- "Eigen" is a German word meaning (roughly) "characteristic".

- The eigenvectors and eigenvalues of a matrix A characterize its behavior.
- An eigenvector is a vector satisfying

$$Av = \lambda v,$$

where λ is a (possibly complex) constant, and $v \neq 0$.

- That is, multiplying by A does nothing to the vector except change its length!
- This is a very unusual vector. There are usually only n of them if A has size $n \times n$.
- Note that if v is an eigenvector, kv is also an eigenvector—so eigenvectors are often normalized to have unit length: $\|v\| = 1$.
- The constant λ is an eigenvalue. Specifically, it is the eigenvalue associated with eigenvector v.
- Since there are (usually) n eigenvectors with n corresponding eigenvalues, we label the eigenvectors and eigenvalues v_i and λ_i where $1 \leq i \leq n$.

Why is this important?

- Suppose we have a vector $x = v_1 + 2v_2$. Then, $Ax = \lambda_1 v_1 + 2\lambda_2 v_2$.
- That is, if we decompose the input into “eigenvector coordinates,” then multiplication by A is simply adding together scaled eigenvectors.
- Assume that v_1, v_2, ... v_n are linearly independent.

$$Av_i = \lambda_i v_i \qquad i = 1, 2, \ldots, n$$

$$A\underbrace{\begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}}_{V} = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}\underbrace{\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}}_{\Lambda}$$

- $AV = V\Lambda$ ➠ $V^{-1}AV = \Lambda$ or $A = V\Lambda V^{-1}$.
- Then, we can write $y = Ax$ as

$$y = V\Lambda V^{-1}x.$$

- V^{-1} decomposes x into the “eigenvector coordinates”. Λ is a diagonal matrix multiplying each component of the resulting vector by the eigenvalue associated with that component, and V puts everything back together.
- Thus, eigenvectors are the “directions” of matrix A, and the eigenvalues are the magnifications along those directions.
- To find eigenvalues, consider that $(\lambda I - A)v = 0$.
- Since $v \neq 0$ $\lambda I - A$ must drop rank for some value of λ associated with v. A matrix which is not full rank has zero determinant. So, we can solve for the eigenvalues by solving

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

- Note that there are very efficient and numerically robust methods of finding eigenvectors and eigenvalues. These methods do not use the determinant rule, above. The determinant rule is useful for mathematical analysis.
- In MATLAB,

```
[V,Lambda] = eig(A);
```