Linear Algebra Review

- Materials from linear algebra review by Prof. Zico Kolter from CMU
- online avainp.linargble
    - http://www.cs.cmu.edu/~zkolter/course/linalg/ (http://www.cs.cmu.edu/~zkolter/course/linalg/)
    - http://www.cs.cmu.edu/~jingx/docs/linearalgebra.pdf (http://www.cs.cmu.edu/~jingx/docs/linearalgebra.pdf)

By Prof. Seungchul Lee
iSystems Design Lab
http://isystems.unist.ac.kr/
UNIST

Table of Contents

- I. 1. Linear Equations
- II. 2. System of Linear Equations
    - I. 2.1. Elements of a Matrix
    - II. 2.2. Vector-Vector Products
    - III. 2.3. Matrix-Vector Products
    - IV. 2.4. Symmetric Matrices
    - V. 2.5. Diagonal Matrices
- III. 3. Norms (strenth or distance in linear space)
    - I. 3.1. Orthogonality
    - II. 3.2. Angle between Vectors

1. Linear Equations

Set of linear equations (two equations, two unknowns)

$$\begin{aligned} 4x_1 - 5x_2 &= -13 \\ -2x_1 + 3x_2 &= 9 \end{aligned}$$

Solving Linear Equations

- Two linear equations

$$\begin{aligned} 4x_1 - 5x_2 &= -13 \\ -2x_1 + 3x_2 &= 9 \end{aligned}$$

- In vector form, $Ax = b$, with

$$A = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} -13 \\ 9 \end{bmatrix}$$

- Solution using inverse

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

- Won’t worry here about how to compute inverse, but it’s very siminp.linargr to the standard method for solving linear equations
- We will use a numpy to compute

In [51]:

```
import numpy as np
```

In [52]:

```
A = np.array([[4, -5],
              [-2, 3]])
b = np.array([[-13],
              [9]])

x = np.linalg.inv(A).dot(b)
x
```

Out[52]:

```
array([[ 3.],
       [ 5.]])
```

2. System of Linear Equations

- consider system of linear equations

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

- can be written in matrix form as $y = Ax$, where

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

2.1. Elements of a Matrix

- Can write a matrix in terms of its columns

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

- Careful, a_i here corresponds to an entire vector $a_i \in \mathbb{R}^m$, not an element of a vector

- Siminp.linargrly, can write a matrix in terms of rows

$$A = \begin{bmatrix} - & b_1^T & - \\ - & b_2^T & - \\ & \vdots & \\ - & b_m^T & - \end{bmatrix}$$

- $b_i \in \mathbb{R}^n$

2.2. Vector-Vector Products

- Inner product: $x, y \in \mathbb{R}^n$

$$x^T y = \sum_{i=1}^{n} x_i \, y_i \qquad \in \mathbb{R}$$

In [53]:

```
x = np.array([[1],
              [1]])
y = np.array([[2],
              [3]])

x.T.dot(y)
```

Out[53]:

```
array([[5]])
```

2.3. Matrix-Vector Products

- $A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n \iff Ax \in \mathbb{R}^m$
- Writing A by rows, each entry of Ax is an *inner product* between x and a row of A

$$A = \begin{bmatrix} - & b_1^T & - \\ - & b_2^T & - \\ & \vdots & \\ - & b_m^T & - \end{bmatrix}, \qquad Ax \in \mathbb{R}^m = \begin{bmatrix} b_1^T x \\ b_2^T x \\ \vdots \\ b_m^T x \end{bmatrix}$$

- Writing A by columns, Ax is a *linear combination of the columns* of A, with coefficients given by x

$$A = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix}, \qquad Ax \in \mathbb{R}^m = \sum_{i=1}^{n} a_i x_i$$

2.4. Symmetric Matrices

- Symmetric matrix:

$$A \in \mathbb{R}^{n \times n} \quad \text{with} \quad A = A^T$$

- Arise naturally in many settings

$$\text{For A} \in \mathbb{R}^{m \times n}, \qquad A^T A \in \mathbb{R}^{n \times m} \ \text{ is symmetric}$$

2.5. Diagonal Matrices

- For $d = [d_1, \cdots, d_n]^T \in \mathbb{R}^n$

$$\operatorname{diag}(d) = \mathbb{R}^{n \times n} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

- For example, the identity is given by $I = \operatorname{diag}(1)$
- Multiplying $A \in \mathbb{R}^{m \times n}$ by a diagonal matrix $D \in \mathbb{R}^{n \times n}$ on the right scales the *columns* of A

$$AD = \begin{bmatrix} | & | & & | \\ d_1 a_1 & d_2 a_2 & \cdots & d_n a_n \\ | & | & & | \end{bmatrix}$$

In [54]:

```
d = np.array([1, 2, 3])
D = np.diag(d)
D
```

Out[54]:

```
array([[1, 0, 0],
       [0, 2, 0],
       [0, 0, 3]])
```

3. Norms (strenth or distance in linear space)

- A vector norm is any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with

    1. $f(x) \geq 0$ and $f(x) = 0 \quad \iff \quad x = 0$
    2. $f(ax) = |a| f(x)$ for $a \in \mathbb{R}$
    3. $f(x + y) \leq f(x) + f(y)$

- l_2 norm

$$\left\Vert x \right\Vert_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

- l_1 norm

$$\left\Vert x \right\Vert_1 = \sum_{i=1}^{n} \left\vert x_i \right\vert$$

- $\lVert x\rVert$ measures length of vector (from origin)

In [55]:

```
x = np.array([[4],
              [3]])

np.linalg.norm(x, 2)
```

Out[55]:

```
5.0
```

In [56]:

```
np.linalg.norm(x, 1)
```

Out[56]:

```
7
```

3.1. Orthogonality

- Two vectors $x, y \in \mathbb{R}^n$ are *orthogonal* if

$$x^T y = 0$$

- They are *orthonormal* if, in addition,

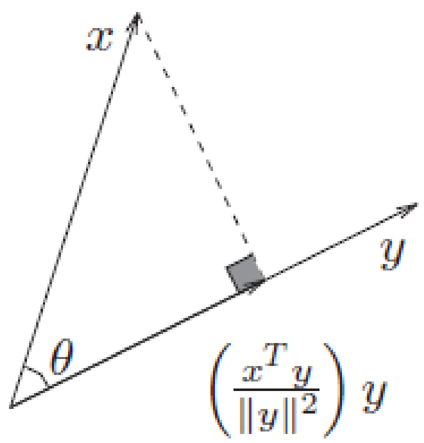
$$\|x\|_2 = \|y\|_2 = 1$$

3.2. Angle between Vectors

- for any $x, y \in \mathbb{R}^n$, $|x^T y| \leq \|x\| \, \|y\|$
- (unsigned) angle between vectors in $\mathbb{R}^n$ defined as

$$\theta = \angle(x, y) = \cos^{-1} \frac{x^T y}{\|x\| \|y\|}$$

$$\text{thus } x^T y = \|x\| \|y\| \cos\theta$$



$\{x \mid x^T y \leq 0\}$ defines a halfspace with outward normal vector y, and boundary passing through 0