Assignment06

November 4, 2018

- Name : Joonyoung-Choi
- Student ID: 20112096
- Description: Least-square solution
- github: https://github.com/mydream757/Computer_Vision

1. Import liabraries

- import needed libraries.

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

2. Initial values

- values about data.

```
In [2]: num     = 201
        std     = 20
```

- values about the line of original fomula.

```
In [3]: a       = 2
        b       = 10
```

3. Generate noisy data

- get (x_i, y_i) data set

```
In [4]: n       = np.random.rand(num)
        nn      = n - np.mean(n)
        x       = np.linspace(-100,100,num)
        y1      = a * x + nn * std + b
```

- this is the expected data from original fomula.

```
In [5]: y2      = a * x + b
```

4. Find least-square solution.

- from data (x_i, y_i)s, we can compute S. $S = \sum (ax_i + b - y_i)^2$

- To find the solution, compute derivatives of a and b.

$$dS/da = 2a\sum x_i^2 + 2b\sum x_i - 2\sum x_i y_i = 0$$
$$dS/db = 2b\sum 1 + 2a\sum x_i - 2\sum y_i = 0$$

- Now, we can find the common solution a, b.

```
In [6]: sumXSquare = np.sum(x**2)
        sumX = np.sum(x)
        sumY = np.sum(y1)
        sumXY = np.sum(x*y1)
        #compute values about approximating line
        pa = (sumXY - (sumX * sumY)/num)/(sumXSquare-(sumX**2)/num)
        print('pa: ',pa)
        pb = -(sumX/num)*pa + sumY/num
        print('pb: ',pb)
```

```
pa:  2.0033756550710997
pb:  10.000000000000009
```

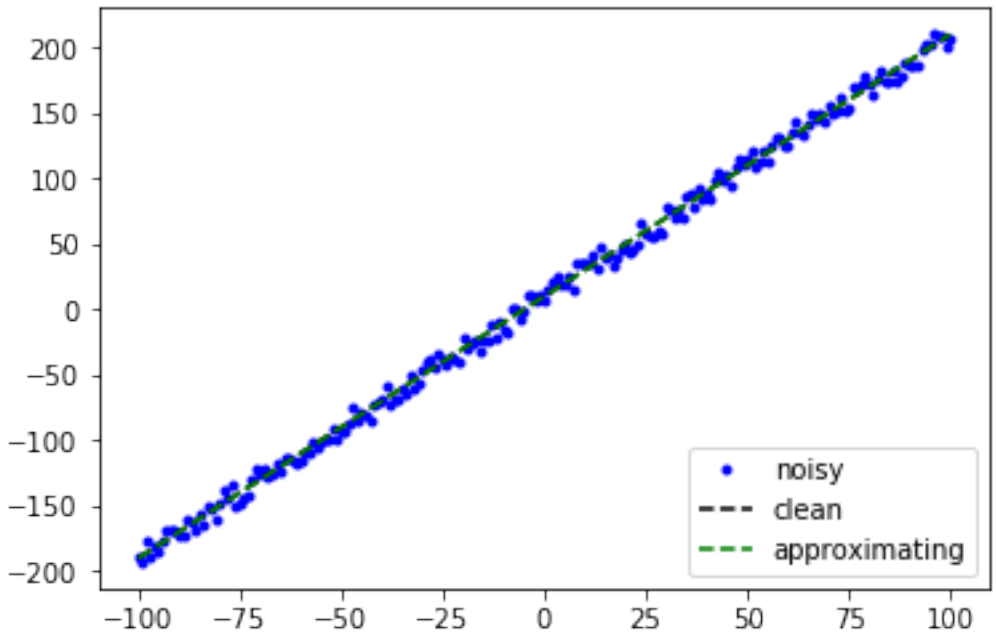
5. Draw the data, original(clean) data and approximating data.

- fomula of approximating line.

```
In [7]: y3 = pa * x + pb
```

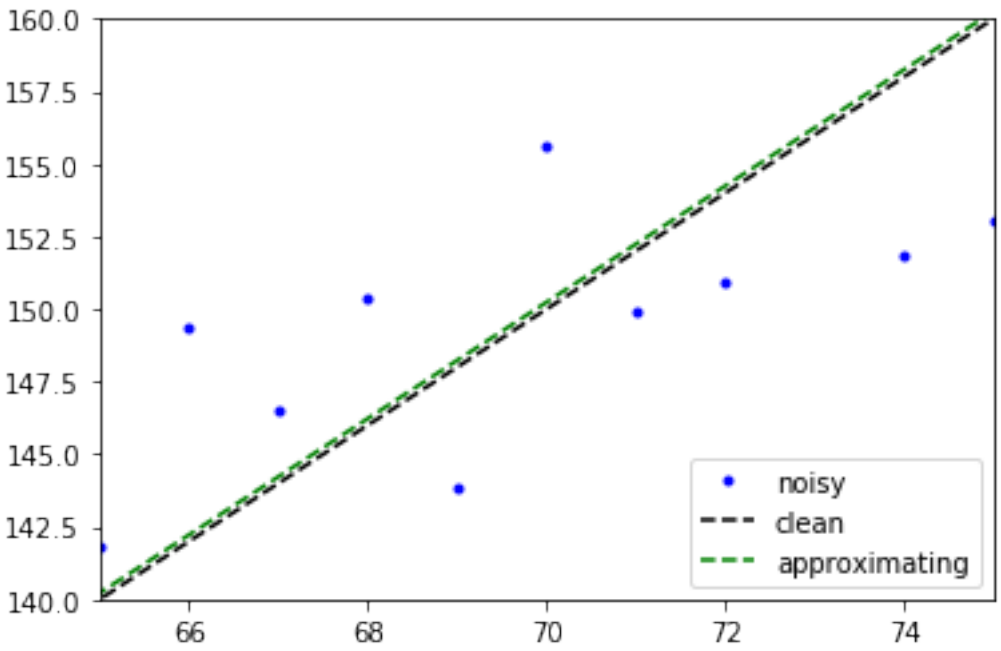
- plot the all

```
In [8]: plt.figure()
        plt.plot(x, y1, 'b.', label='noisy')
        plt.plot(x, y2, 'k--', label='clean')
        plt.plot(x, y3, 'g--',label='approximating')
        plt.legend(loc='lower right')
        plt.show()
```



- redraw the plot to show the difference between the lines.

```
In [9]: plt.figure()
        plt.plot(x, y1, 'b.', label='noisy')
        plt.plot(x, y2, 'k--', label='clean')
        plt.plot(x, y3, 'g--',label='approximating')
        plt.legend(loc='lower right')
        plt.xlim(65,75)
        plt.ylim(65*a+b,75*a+b)
        plt.show()
```



- we can see that the difference between clean data and approximating data from noisy.