Pattern Recognition Assignment 02

March 20, 2019

20145271

0.1 Taylor Approximation

Compare the original function with Taylor Approximation

1) Origin function

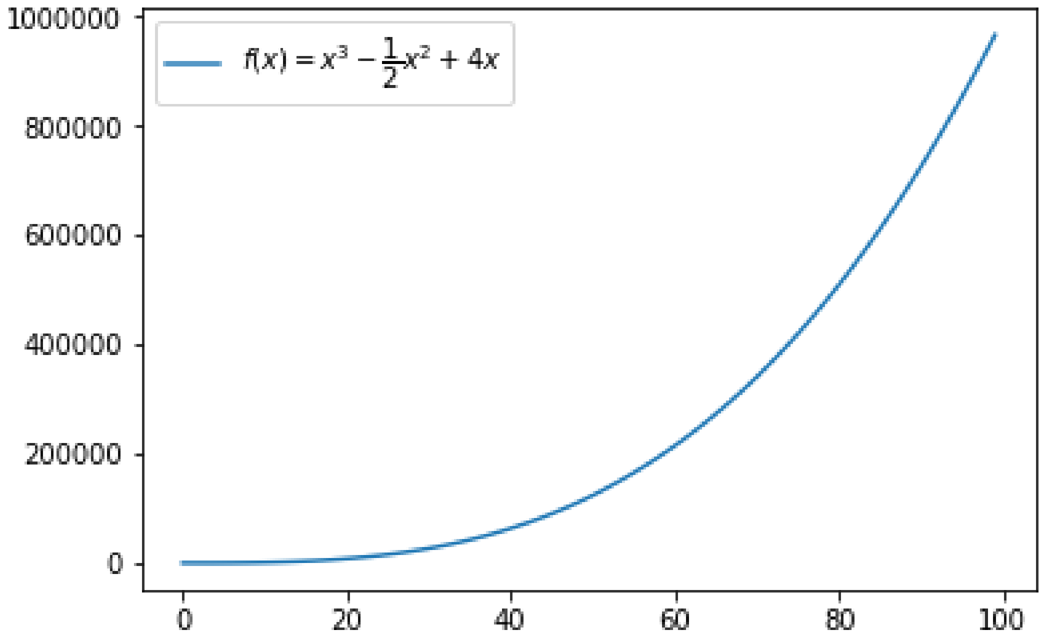
$$f(x) = x^3 - \frac{1}{2}x^2 + 4x \quad (0 \le x \le 100)$$

2) Plot origin function $f(x)$

```
In [107]: %matplotlib inline

          import numpy as np
          import matplotlib.pyplot as plt

          #x = np.linspace(0, 15, 1024)
          x = [a for a in range(0, 100)]
          y = [b*b*b - 1/2*b*b + 4*b for b in range(0, 100)]
          plt.plot(x, y, label = r'$f(x)=x^3 - \dfrac{1}{2}x^2 + 4x$')
          plt.legend()
          plt.show()
```



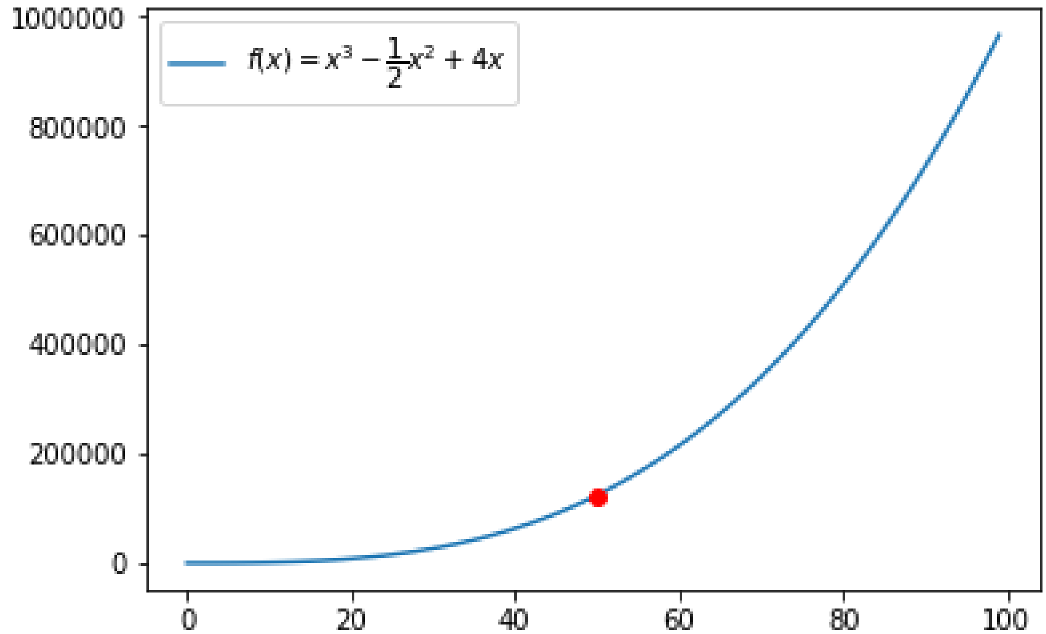
3) Select a point A(50, 123950) on function $f(x)$

```
In [101]: %matplotlib inline

          import numpy as np
          import matplotlib.pyplot as plt

          #x = np.linspace(0, 15, 1024)
          x = [a for a in range(0, 100)]
          y = [b*b*b - 1/2*b*b + 4*b for b in range(0, 100)]

          y2 = [7454*x - 248750 for x in range(0,100)]
          plt.plot(x, y, label = r'$f(x)=x^3 - \dfrac{1}{2}x^2 + 4x$')
          plt.plot(50,123950, 'ro--')
          plt.legend()
          plt.show()
```



4) First-order Taylor approximation at the Point A

In function f(x), we can approximate f(x) about the point where x = a by the polynomial

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

If we want to approximate this to first order, it means

$$\hat{f}(x) \approx f(a) + f'(a)(x-a)$$

In this case, The x coordinates of Point A is 50. (x = 50)

$$\hat{f}(x) \approx f(50) + f'(50)(x-50) f'(x) = \frac{\partial f}{\partial x}(x^3 - \frac{1}{2}x^2 + 4x) = 3x^2 - x + 4 f'(50) = 7454 \therefore \hat{f}(x) \approx 7454x - 247850$$

5) Plot $f(x)$ and $\hat{f}(x)$ on Point A

```
In [106]: %matplotlib inline

          import numpy as np
          import matplotlib.pyplot as plt

          #x = np.linspace(0, 15, 1024)
          x = [a for a in range(0, 100)]
          y = [b*b*b - 1/2*b*b + 4*b for b in range(0, 100)]
```

```
y2 = [7454*x - 248750 for x in range(0,100)]
plt.plot(x, y, label = r'$f(x)=x^3 - \dfrac{1}{2}x^2 + 4x$')
plt.plot(x, y2, label = r'$\hat{f}(x)=3x^2 - x + 4$')
plt.plot(50,123950, 'ro--')
plt.legend()
plt.show()
```