Example 2

January 25, 2017

0.1 With a non-uniform distribution (Max-Lloyd Quantizer)

Laplacian pdf: ####

$$p(x) = e^{-0.5\cdot|x|}$$

1) Random initialization:

$$y_1 = 0.3, y_2 = 0.8$$

2) Nearest neighbour:

$$b_1 = (0.3 + 0.8)/2 = 0.55$$

3) Conditional expectation:

$$y_k = \frac{\int\limits_{b_{k-1}}^{b_k} x\cdot(x)dx}{\int\limits_{b_{k-1}}^{b_k} p(x)dx}$$

Now we need Python to compute the numerator integral, for y_1,

$$\int\limits_0^{b_1} x\cdot(x)dx = \int\limits_0^{0.55} x.e^{-0.5\cdot|(x)|}dx$$

In Python we can use the function "scipy.integrate.quad" for integration (type "help(quad)" to get information about its use)

```
In [1]: from scipy.integrate import quad
        import numpy as np

        Num, Nerr = quad(lambda x:x * np.exp(-0.5 * np.abs(x)), 0, 0.55)
        print Num
```

```
0.126182171553
```

For the denominator integral we get,

$$\int\limits_0^{0.55} p(x)dx$$

```
In [2]: Den, Derr = quad(lambda x: np.exp(-0.5*np.abs(x)), 0, 0.55)
        print Den
```

```
0.48085575355
```

```
In [3]: print Num/Den
```

```
0.262411691284
```

and hence we obtain,

$$y_1 = \frac{Num}{Den} = \frac{0.12618}{0.48086} = 0.2624$$

For y_2 we get,

```
In [4]: Num, Nerr = quad(lambda x:x * np.exp(-0.5 * np.abs(x)), 0.55, 1)
        print Num
```

```
0.234633870172
```

```
In [5]: Den, Derr = quad(lambda x: np.exp(-0.5*np.abs(x)), 0.55, 1)
        print Den
```

```
0.306082927025
```

```
In [6]: print Num/Den
```

```
0.76656961057
```

0.1.1 Hence $y_2 = 0.76657$.

Go back from here to step 2(compute nearest neighbor) until convergence.