hw3

October 21, 2018

In [85]:
```
import numpy as np
import random
import math
import matplotlib.pyplot as plt
```

In [113]:
```
vector_2 = [0,1,0,0,0,0,0,0,0,0]
vector_4 = [0,0,0,1,0,0,0,0,0,0]
vector_6 = [0,0,0,0,0,1,0,0,0,0]
vector_8 = [0,0,0,0,0,0,0,1,0,0]
vector_10 = [0,0,0,0,0,0,0,0,0,1]
vectors = [vector_2,vector_4,vector_6,vector_8,vector_10]
fb = [1,2,4,8,16,32,64,128,256,512]
graph_start = 50000
graph_step = 50000
graph_end = 3000000
num_points = (graph_end - graph_start)/graph_step

plt.figure(figsize = [15,15])

for i in range(0,5):
    sample_index = 0
    probs = np.zeros(int(num_points))
    for num_samples in np.arange(graph_start,graph_end,graph_step):
        rand_samples = np.random.rand(num_samples,10)
        samples = rand_samples>0.5
        result = np.dot(samples,vectors[i])
        fb_result = np.dot(samples,fb)
        pz_given_b = (2/3*np.power(0.2,np.abs(fb_result-128)))
        numerator = np.dot(result,pz_given_b)
        denominator = np.sum(pz_given_b)
        pb_given_z = numerator/denominator
        probs[sample_index] =  pb_given_z
        sample_index +=1

    plt.subplot(5, 1, i+1)
    plt.plot(np.arange(graph_start,graph_end,graph_step),probs)
    plt.xlabel('number of samples')
    plot_index = (i+1)*2
```

```
        plt.ylabel('P(B%d) given Z = 128' %plot_index)
        plt.title('Posterior probability of B%d' %plot_index)

    plt.tight_layout()
    plt.show()
```