7_3

May 2, 2016

```
In [3]: import numpy as np
        import scipy.stats as stats
        %matplotlib inline
        import matplotlib.pyplot as plt
```

```
In [6]: sample = stats.cauchy.rvs(size=100)
```

Очевидно, что $a = 0$. Найдем σ, учитывая, что $P(-0.5 < \theta < 0.5) = 0.95$. В силу симметрии $N(0, \sigma^2)$, достаточно найти квантиль от $0.5 - 0.5 * p$, где $p = 0.95$

```
In [85]: z = stats.norm.ppf(0.5 - 0.5 * 0.95)
         sigma = abs(1 / (2 * z))
         print(sigma)
```

```
0.255106728462
```

Байесовская оценка $\hat{\theta_n} = \frac{\sigma^2 \cdot n \cdot \overline{X} + a}{\sigma^2 \cdot n + 1}$

```
In [88]: def bayers_est(sample, a, sigma):
             n = len(sample)
             sigmaSq = sigma ** 2
             avg = np.average(sample)
             return (sigmaSq * n * avg + a) / (sigmaSq * n + 1)
```

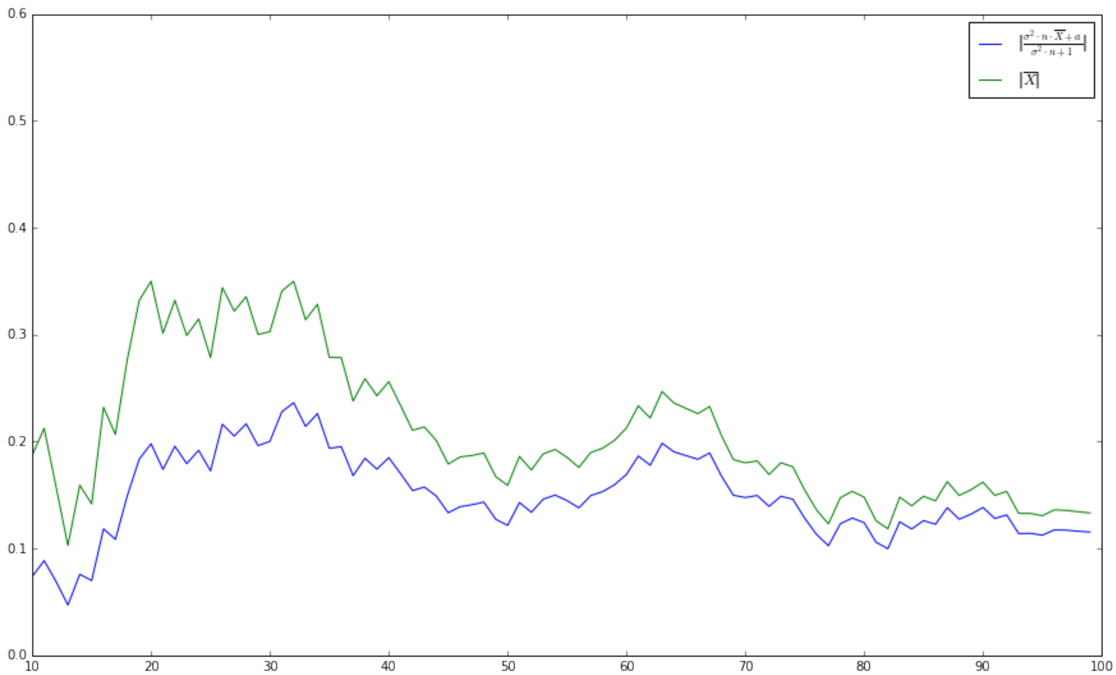
Оценка максимального правдоподобия $\hat{\theta_n} = \overline{X}$

```
In [89]: def MLE_est(sample):
             return np.average(sample)
```

```
In [130]: N = 100
          def draw_plot():
              sample = stats.norm.rvs(size=N)
              grid = np.arange(1, 100, 1)

              plt.figure(figsize=(15,9))
              plt.plot(grid, np.abs([bayers_est(sample[:n], 0, sigma) for n in grid]), label=r'|$\frac{\
              plt.plot(grid, np.abs([MLE_est(sample[:n]) for n in grid]), label=r'|$\overline{X}$|')
          #     plt.plot(grid, data_unfair, label=r'|$\hat{\theta}_{Beta(0.5, 0.5)} - \theta$|')
          #     plt.plot(grid, data_CoA_unfair, label=r'|$\hat{\theta}_{Beta(10, 0.5)} - \theta$|')
          #     plt.plot(grid, data_mle, label=r'$|\theta - \overline{X}$|')
          #     plt.legend(loc='best')

          #     plt.xlim(1, 20)
              plt.xlim(10, 100)
              plt.ylim(0, 0.6)
```

```
#     plt.ylim(-0.1, 0.3)
    plt.legend(loc='best')


    plt.show()
```

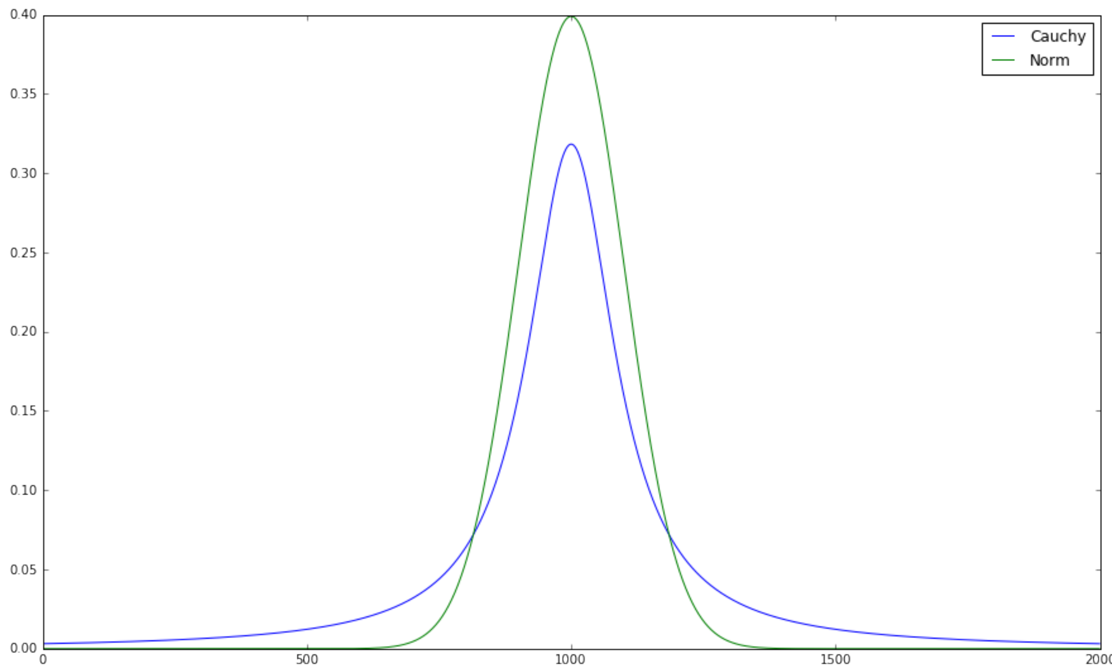
In [132]:
```
draw_plot()
```



Приближение распределения Коши нормальным распределением связано с тем, что у них довольно похожие плотности распределения:

In [135]:
```
plt.figure(figsize=(15,9))
grid = np.arange(-10, 10, 0.01)
plt.plot(stats.cauchy.pdf(grid), label=r'Cauchy')
plt.plot(stats.norm.pdf(grid), label=r'Norm')
plt.legend(loc='best')
```

Out[135]:
```
<matplotlib.legend.Legend at 0x10c6f6a58>
```



Хотя, в действительности у распределения Коши даже мат. ожидания нет, но приближая его стандартным нормальным можно получить оценку параметра.

```
In [ ]:
```