In [117]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
import math
%matplotlib inline
```

Сгенерируйте выборку $X_1, \ldots, X_{100}$ из распределения $N(0, 1)$. Для каждого n <= 100 в модели $N(\theta, 1)$ найдите оценку максимального правдоподобия по выборке $X_1, \ldots, X_n$ и байесовскую оценку, для которой в качестве априорного распределения возьмите сопряженное из теоретической задачи.

In [134]:
```
big_sample = sts.norm(0., 1.).rvs(size=100)
```

Найдем для $X_1, \ldots, X_n$ в модели $N(\theta, 1)$ оценку максимального правдоподобия:

$$f_\theta(X) = \rho_\theta(X_1) \cdot \ldots \cdot \rho_\theta(X_n) = (\frac{1}{\sqrt{2\pi}})^n e^{-\frac{1}{2}\sum(X_i-\theta)^2} \rightarrow max$$

$$L_\theta = lnf_\theta = \ldots = \theta \sum X_i - \frac{n}{2}\theta^2 \rightarrow max$$

$$L'_\theta = \sum X_i - n\theta = 0 \rightarrow \theta^* = \frac{\sum X_i}{n}$$

Сопряженное распределение:

N($\left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^n x_i}{\sigma^2}\right) \Big/ \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right), \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1}$)

Нам нужно найти байесовскую оценку матожидания θ,

то есть условное матожидание сопряженного распределения $E(\theta|X)$, которое как раз равно

$$\left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^n x_i}{\sigma^2}\right) \Big/ \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right) = \theta^*$$

Знаем, что σ^2 = 1 :

$$\theta^* = \left(\frac{\mu_0}{\sigma_0^2} + \sum_{i=1}^n x_i\right) \Big/ \left(\frac{1}{\sigma_0^2} + n\right)$$

In [135]:
```
def bayes_estimator(X_array, param):
    E, D = param[0], param[1]
    return (E/D + X_array.sum()*1.) / (1./D + len(X_array)*1.)

def likelihood_estimator(X_array):
    return X_array.mean()
```

Возьмите несколько значений параметров сдвига и масштаба для априорного распределения (μ_0, σ_0^2):

1) (0, 1),

2) (0, 100),

3) (10, 1),

4) (10, 100).

Постройте графики абсолютной величины отклонения оценки от истинного значения параметра в зависимости от n для оценки максимального правдоподобия и байесовских оценок, которым соответствуют разные значения параметров априорного распределения (5 кривых на одном графике). Сделайте выводы.

Считаем, что при указании параметров распределений в задании используется синтаксис (μ, σ^2)

$N(\theta, 1)$

In [136]:
```
params = [ [0.,1.], [0.,100.], [10.,1.], [10.,100.] ]

sample = np.array([])
bayes_values = [[],[],[],[]]
likelihood_values = []

for n in range(0, len(big_sample)):
    sample = np.append(sample, big_sample[n])
    for i, param in enumerate(params):
        bayes_values[i].append(abs(bayes_estimator(sample, param)))
    likelihood_values.append(abs(likelihood_estimator(sample)))
```

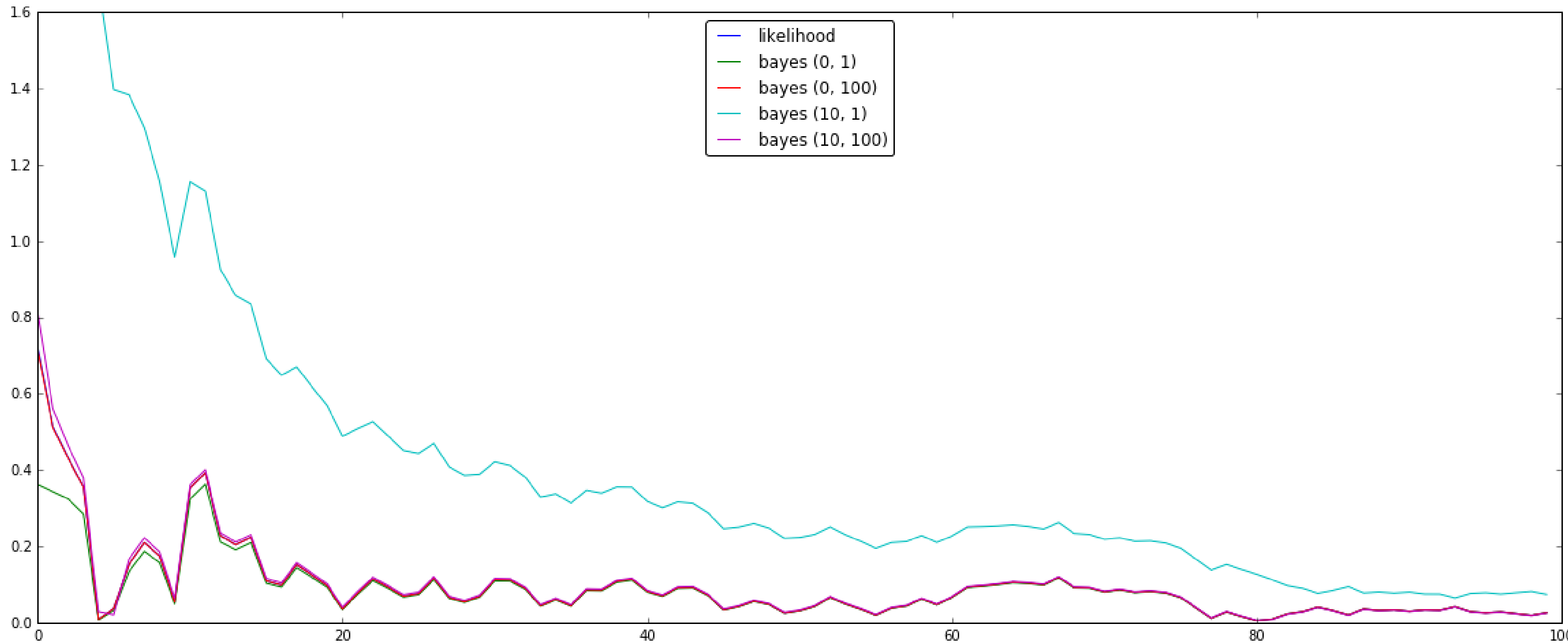
In [137]:
```
N = range(0, len(big_sample))

plt.figure(figsize=(20,8))
plt.ylim(0, 1.6)

plt.plot(N, likelihood_values, label='likelihood')

for i, param in enumerate(params):
    label = "bayes (%d, %d)" % (int(param[0]), int(param[1]))
    plt.plot(N, bayes_values[i], label=label)

plt.legend(loc='upper center', fancybox=True)
plt.show()
```



$N(0, \theta)$

Аналогичные исследования произведите для модели $N(0, \theta)$.

Найдем для $X_1, \ldots, X_n$ в модели $N(0, \theta)$ оценку максимального правдоподобия:

$$f_\theta(X) = \rho_\theta(X_1) \cdot \ldots \cdot \rho_\theta(X_n) = (\frac{1}{\sqrt{2\pi}})^n (\frac{1}{\sigma})^n e^{-\frac{1}{2\sigma^2}\sum X_i^2} \to max$$

$$L_\theta = lnf_\theta = \ldots = const_\theta - nln\sigma - \frac{\sum X_i^2}{2\sigma^2} \to max$$

$$L_\theta' = \frac{1}{\sigma^3}\sum X_i^2 - \frac{n}{\sigma} = 0 \to \theta^* = \sigma^2 = \frac{\sum X_i^2}{n}$$

Сопряженное распределение:

При известном $\mu = 0$ сопряженным распределением будет обратное гамма-распределение с параметрами

$$\alpha + \frac{n}{2}, \beta + \frac{\sum_{i=1}^n x_i^2}{2}$$

Матожидание, то есть байесовская оценка σ^2, равно

$$E = \frac{\beta'}{\alpha' - 1}$$

$$\frac{\beta + \frac{\sum_{i=1}^n x_i^2}{2}}{\alpha + \frac{n}{2} - 1} = \theta^*$$

In [138]:
```
def bayes_estimator_sigma(X_array, param):
    a, b = param[0], param[1]
    return (b + 0.5*(X_array*X_array).sum()*1.)/(a + 0.5*len(X_array) - 1.)

def likelihood_estimator_sigma(X_array):
    return (X_array*X_array).mean()
```

В этом случае возьмите следующие параметры сдвига и масштаба для априорного распределения (α, β):

1) (1, 1),

2) (1, 100),

3) (10, 1),

4) (10, 100).

In [139]:
```
params = [ [1.,1.], [1.,100.], [10.,1.], [10.,100.] ]

sample = np.array([])
bayes_values = [[],[],[],[]]
likelihood_values = []

for n in range(0, len(big_sample)):
    sample = np.append(sample, big_sample[n])
    for i, param in enumerate(params):
        bayes_values[i].append(abs(1.-bayes_estimator_sigma(sample, param)))
    likelihood_values.append(abs(1.-likelihood_estimator_sigma(sample)))
```

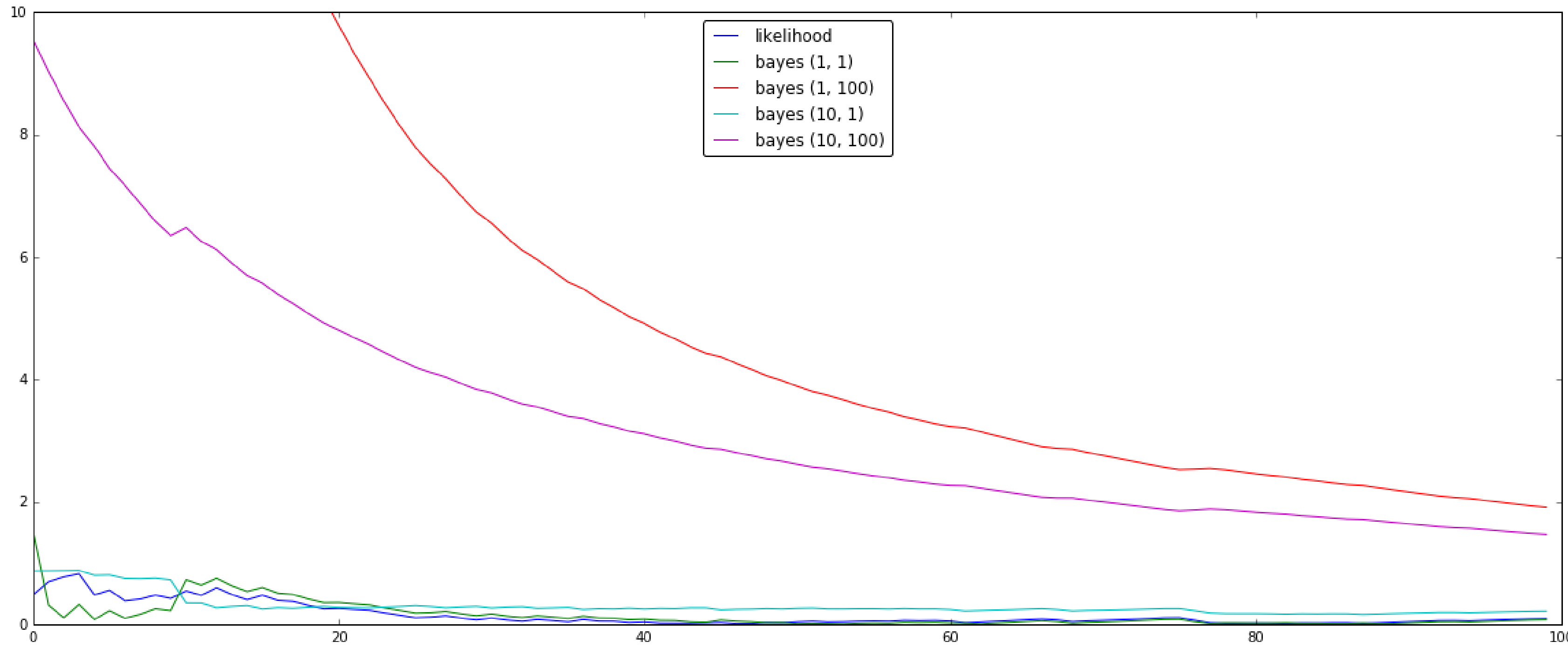
In [140]:
```
N = range(0, len(big_sample))

plt.figure(figsize=(20,8))
plt.ylim(0, 10.)

plt.plot(N, likelihood_values, label='likelihood')

for i, param in enumerate(params):
    label = "bayes (%d, %d)" % (int(param[0]), int(param[1]))
    plt.plot(N, bayes_values[i], label=label)

plt.legend(loc='upper center', fancybox=True)
plt.show()
```



Вывод:

Байесовская оценка очень похожа на оценку максимального правдоподобия (если брать разумные параметры априорного распределения), а иногда лучше неё.

Самое главное - использовать догадки, каким может быть параметр, подбирая параметры байеса

In [ ]:

```
```