Machine Learning
MSE FTP MachLe
Christoph Würsch

Bayesian Inference for a Gaussian

Extended Solution for A10, Series 8

- Author: Christoph Würsch, OST
- MSE FTP_MachLE

Assume a Gaussian likelihood function of the follwing form with known variance σ.

$$p(\mathbf{X} \mid \mu) = \prod_{n=1}^{N} p(x_n \mid \mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}$$

In this case, the posterior probability distribution will again be a Gaussian distribution $\mathcal{N}$ and has the same form as the prior. The prior is then called a *conjugate* prior to the posterior.

Again we emphasize that the likelihood function $p(\mathbf{X} \mid \mu)$ is not a probability distribution over μ and is not normalized. We see that the likelihood function takes the form of the exponential of a quadratic form in μ. Thus if we choose a prior $p(\mu)$ given by a Gaussian, it will be a *conjugate* distribution for this likelihood function because the corresponding posterior will be a product of two exponentials of quadratic functions of μ and hence will also be Gaussian $\mathcal{N}$. We therefore take our prior distribution to be

$$p(\mu) = \mathcal{N}\left(\mu \mid \mu_0, \sigma_0^2\right)$$

And the posterior distribution is given by:

$$p(\mu \mid \mathbf{X}) \propto p(\mathbf{X} \mid \mu) \cdot p(\mu)$$

We consider only the terms in the exponentials and neglect the normalization factor. In this case, we have:

$$\exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \cdot \exp\left\{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2\right\}$$

Now, we only look at the quadratic term Q. The posterior probability distribution is a function of μ. So we are interested only in the linear and quadratic terms in μ.

$$
\begin{aligned}
Q =& \left\{ -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{1}{2\sigma_0^2}(\mu-\mu_0)^2 \right\} \\
=& \left\{ -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n^2-2x_n\mu+\mu^2) - \frac{1}{2\sigma_0^2}(\mu^2-2\mu\mu_0+\mu_0^2) \right\} \\
=& \left\{ -\frac{1}{2\sigma^2}\left(\sum_{n=1}^{N}x_n^2-2\mu\sum_{n=1}^{N}x_n+N\mu^2\right) - \frac{1}{2\sigma_0^2}(\mu^2-2\mu\mu_0+\mu_0^2) \right\} \\
=& \left\{ -\mu^2\left(\frac{1}{2\sigma_0^2}+\frac{N}{2\sigma^2}\right) + 2\mu\left(\frac{\sum_{n=1}^{N}x_n}{2\sigma^2}+\frac{\mu_0}{2\sigma_0^2}\right) + \dots \right\} - \frac{1}{2}\left\{ \left(\frac{1}{\sigma_0^2}+\frac{N}{\sigma^2}\right)\left[\mu - \frac{\frac{\sum_n}{\sigma^2}}{\frac{1}{\sigma_0^2}}\right.\right.
\end{aligned}
$$

The term in front of the square bracket is the inverse variance:

$$
\frac{1}{\sigma_N^2} = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2} \tag{1}
$$

The shift term in the quare bracket is the mean μ_N:

$$
\begin{aligned}
\mu_N &= \frac{\frac{\sum_n x_n}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}} \\
&= \frac{\sigma_0^2\sum_n x_n + \sigma_0^2\mu_0}{\sigma^2 + N\sigma_0^2} \\
&= \frac{\sigma^2}{\sigma^2+N\sigma_0^2}\cdot\mu_0 + \frac{N\sigma_0^2}{\sigma^2+N\sigma_0^2}\cdot\frac{1}{N}\sum_n x_n \\
&= \frac{\sigma^2}{\sigma^2+N\sigma_0^2}\cdot\mu_0 + \frac{N\sigma_0^2}{\sigma^2+N\sigma_0^2}\cdot\mu_{\mathrm{ML}}
\end{aligned}
$$

It is worth spending a moment studying the form of the posterior mean and variance. Using simple manipulation involving completing the square in the exponent, you could show that the posterior distribution is given by:

$$
p(\mu \mid \mathbf{X}) = \mathcal{N}\left(\mu \mid \mu_N, \sigma_N^2\right) \tag{2}
$$

$$
\frac{1}{\sigma_N^2} = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2} \tag{3}
$$

$$
\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\cdot\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\cdot\mu_{\mathrm{ML}} \tag{4}
$$

in which μ_{ML} is the maximum likelihood solution for μ given by the sample mean

$$
\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n \tag{5}
$$

Python example for a Bayesian Update

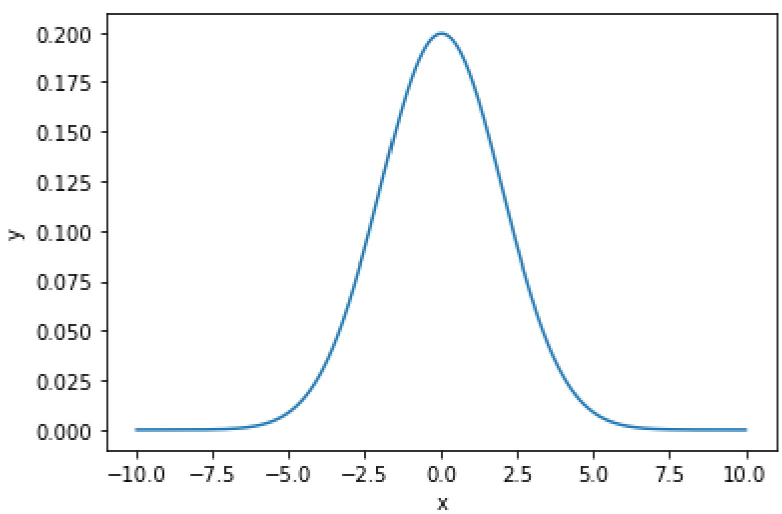
Let's do a concrete example using Python to demonstrate this. We cose a simple Gaussian prior for μ with $\mu_0 = 0$ and $\sigma_0^2 = 4$.

$$p(\mu) = \mathcal{N}(\mu \mid 0, 4) \tag{6}$$

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

mu0=0.0
sigma0=1.0

def NormalGauss(x,mu0=mu0,sigma0=sigma0):
    y = 1/np.sqrt(2*np.pi*sigma0)*np.exp(-1/(2*sigma0)*(x-mu0)**2)
    return y
```

(i) The prior $p(\mu)$ belief about the mean of the data x_i

In [2]:
```
x=np.linspace(-10,10,1000)
prior=NormalGauss(x, mu0=0, sigma0=4)
plt.figure()
plt.plot(x,prior)
plt.set_grid=True
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



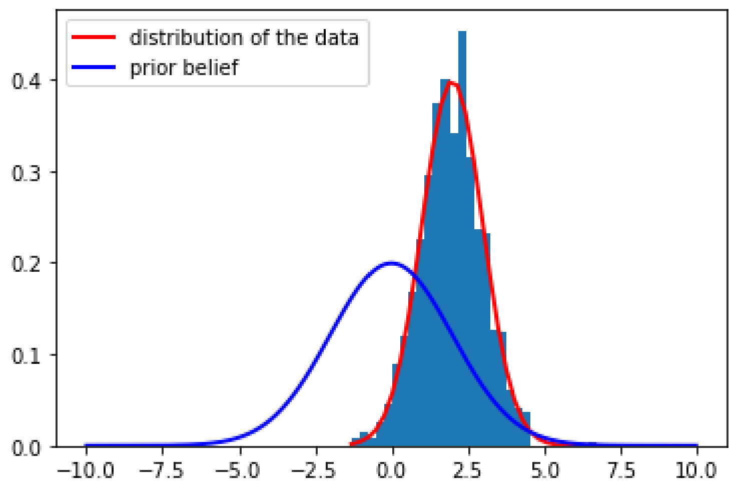
(ii) The distribution of the data x_i

In our model we assume that the distribution of the data is also a Gaussian distribution. Therefore we can caluclate the likelihood of the data x_i given the model and estimate the maximum likelihood mean μ_{ML} and σ^2_{ML}.

Now, lets draw $N = 30$ samples from another Gaussian distribution that describes the distribution of the samples. These will be our measurement points x_i. We assume that the mean and the variance of the data distribution is given by:

$$\mu = 2$$
$$\sigma^2 = 1$$

In [3]:
```
mu, sigma = 2, 1 # mean and standard deviation

#number of samples
N=30

s = np.random.normal(mu, sigma, 1000)

count, bins, ignored = plt.hist(s, N, density=True)
Data=NormalGauss(bins, mu0=mu, sigma0=sigma)

plt.plot(bins, Data ,linewidth=2, color='r',label='distribution of the data')
plt.plot(x,prior,linewidth=2, color='b',label='prior belief')
plt.legend()
plt.show()
```



We calculate the empirical mean and variance of the data x_i.

In [4]:
```
mu_ML=np.mean(s)
var_ML=np.var(s)

print('mean: ',mu_ML)
print('variance: ',var_ML)
```

```
mean:  1.9775767874121195
variance:  1.0319758765876823
```

(iii) Calculation of the posterior distribution

According to our calculations, we have for the posterior:

$$p(\mu \mid \mathbf{X}) = \mathcal{N}\left(\mu \mid \mu_N, \sigma_N^2\right) \tag{7}$$

$$\frac{1}{\sigma_N^2} = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2} \tag{8}$$

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2} \cdot \mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2} \cdot \mu_{\mathrm{ML}} \tag{9}$$

in which μ_{ML} is the maximum likelihood solution for μ given by the sample mean

$$\mu_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{10}$$

In [5]:
```
sigma_N=1/(N/var_ML+1/sigma0)
print('sigma_N^2:', sigma_N)

mu_N=sigma/(N*sigma0+sigma)*mu0+N*sigma0/(N*sigma0+sigma)*mu_ML
print('mu_N:', mu_N)
```

```
sigma_N^2: 0.03325524229239507
mu_N: 1.9137839878181802
```

First of all, we note that the **mean of the posterior distribution** μ_N given by the equation above is a compromise between the prior mean μ_0 and the maximum likelihood solution μ_{ML}. If the number of observed data points $N = 0$, then this equation reduces to the prior mean as expected. For $N \to \infty$, the posterior mean is given by the maximum likelihood solution.

Similarly, consider the result for the **variance of the posterior distribution**. We see that this is most naturally expressed in terms of the inverse variance, which is called the precision. Furthermore, the precisions are additive, so that the precision of the posterior is given by the precision of the prior plus one contribution of the data precision from each of the observed data points. As we increase the number of observed data points, the precision steadily increases, corresponding to a posterior distribution with steadily decreasing variance. With no observed data points, we have the prior variance, whereas if the number of data points $N \to \infty$, the variance σ_N^2 goes to zero and the posterior distribution becomes infinitely peaked around the maximum likelihood solution.

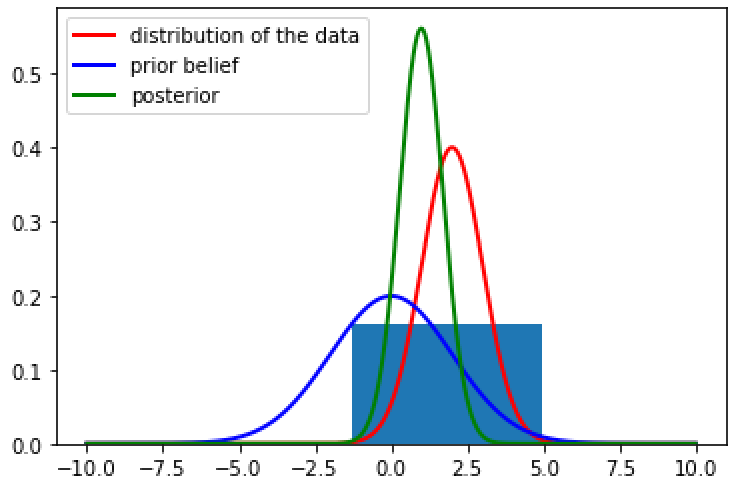
We therefore see that the maximum likelihood result of a point estimate for μ given by μ_{ML} is recovered precisely from the Bayesian formalism in the limit of an infinite number of observations. Note also that for finite N, if we take the limit $\sigma_0^2 \to \infty$ which the prior has infinite variance then the posterior mean μ reduces to the maximum likelihood result, while the posterior variance is given by $\sigma_N^2 = \frac{\sigma^2}{N}$.

Now lets change the number of *measurement points* N contiunously and draw the different distributions:

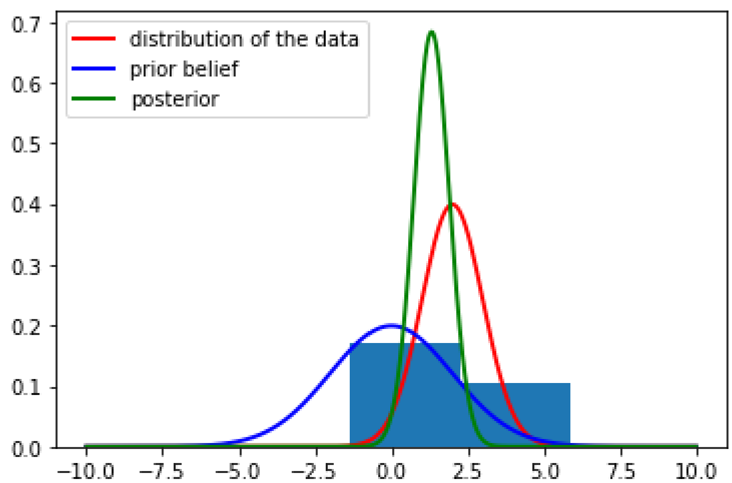
In [6]:
```
#number of samples
NList=[1,2,4,8,12,16,20,80]
x=np.linspace(-10,10,1000)

for N in NList:

    s = np.random.normal(mu, sigma, 1000)

    count, bins, ignored = plt.hist(s, N, density=True)
    Data=NormalGauss(x, mu0=mu, sigma0=sigma)
    sigma_N=1/(N/var_ML+1/sigma0)
    print('sigma_N^2:', sigma_N)
```

```
    mu_N=sigma/(N*sigma0+sigma)*mu0+N*sigma0/(N*sigma0+sigma)*mu_ML
    print('mu_N:', mu_N)

    plt.plot(x, Data ,linewidth=2, color='r',label='distribution of the data')
    plt.plot(x,prior,linewidth=2, color='b',label='prior belief')

    Posterior=NormalGauss(x, mu0=mu_N, sigma0=sigma_N)
    plt.plot(x,Posterior,linewidth=2, color='g',label='posterior')
    plt.legend()
    plt.show()
```

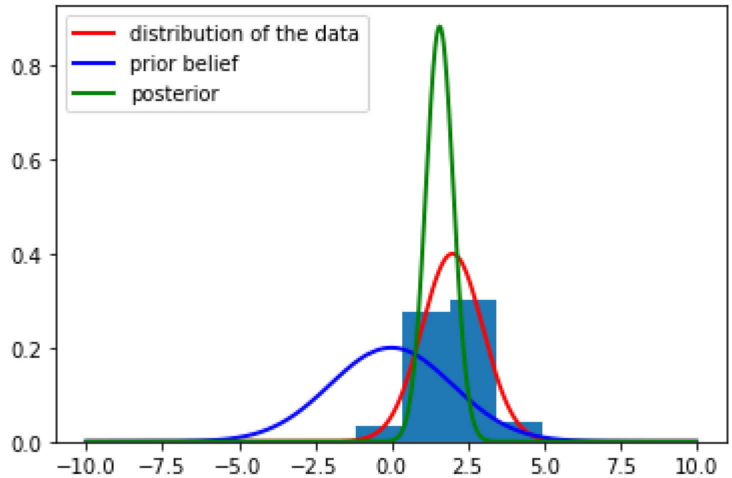
```
sigma_N^2: 0.5078681732780657
mu_N: 0.9887883937060598
```



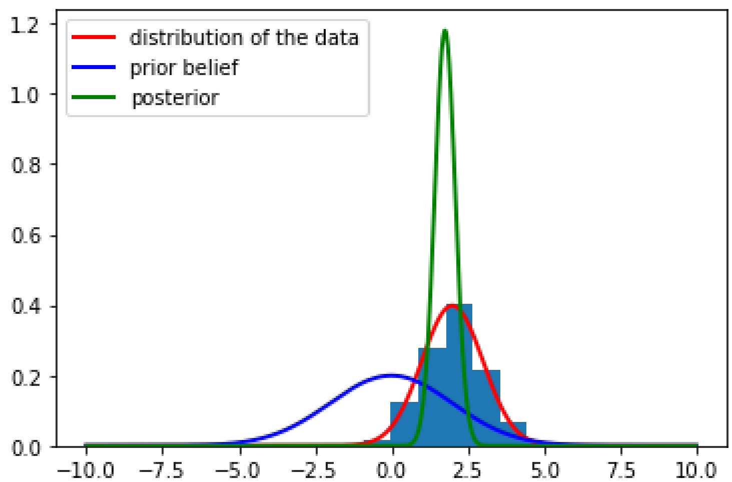

```
sigma_N^2: 0.3403641448985118
mu_N: 1.318384524941413
```



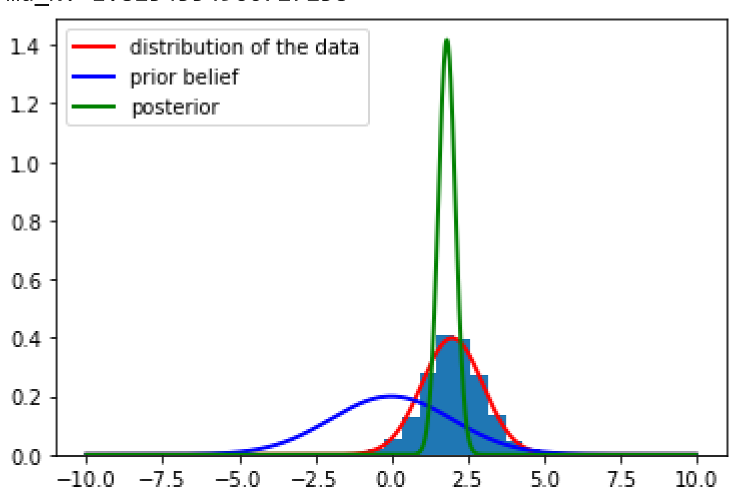

```
sigma_N^2: 0.20508362955179602
mu_N: 1.5820614299296958
```



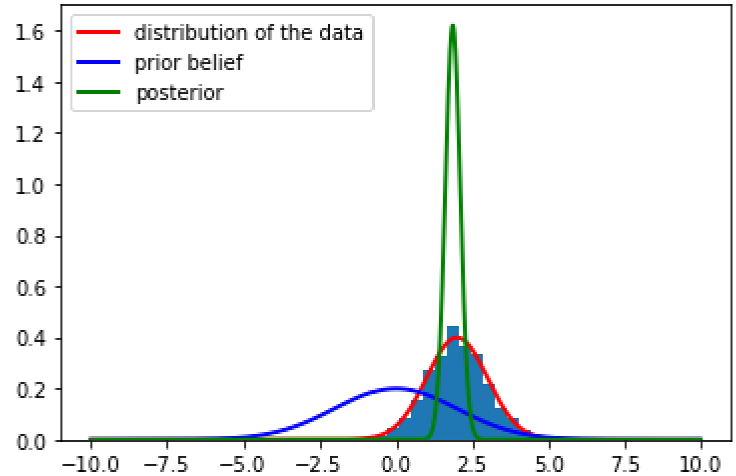

```
sigma_N^2: 0.114258041727362
mu_N: 1.7578460332552173
```



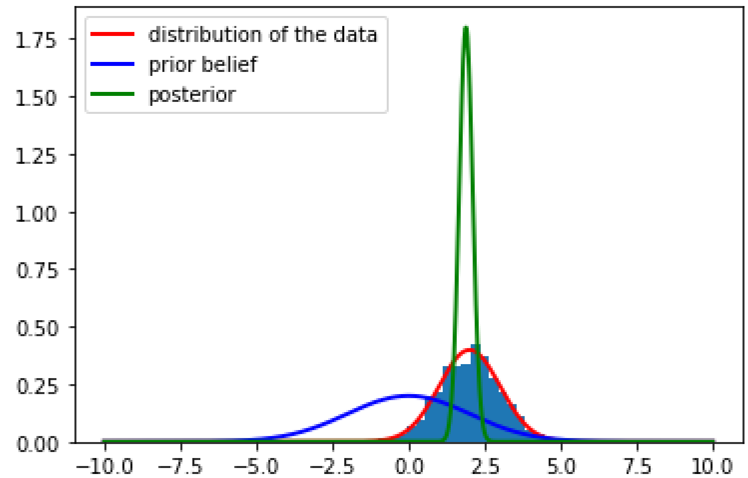

```
sigma_N^2: 0.0791879824180504
mu_N: 1.8254554960727258
```



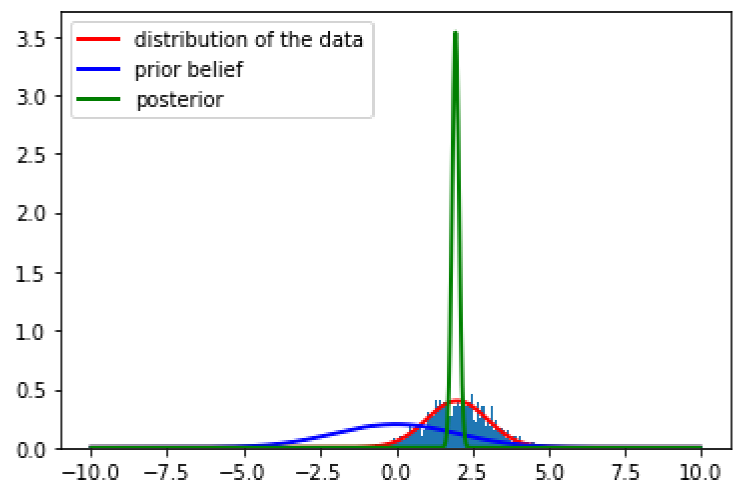

```
sigma_N^2: 0.06059049660857295
mu_N: 1.8612487410937595
```




```
sigma_N^2: 0.04906699601802294
mu_N: 1.8834064642020185
```




```
sigma_N^2: 0.012735415438460854
mu_N: 1.9531622591724636
```



What we observe is, that the posterior distribution gets more and more peaked around the data distribution and the estimate of the true mean μ_N gets very narrowly centered around the data mean. The more data we have, the less important our intial estimate about the prior distribution becomes.

In [ ]: