maximum_likelihood_estimation

March 12, 2021

1 Maximum likelihood estimation

@Author: Javier Cebrián

@Attention: In this file there are Plotly (rendered with HTML) plots and equations. Please, if you are checking it from github use nbviewer.

[9]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import plotly.offline as pyo
from scipy import stats
import plotly.io as pio
pio.renderers.default = "notebook+pdf"
```

1.1 1- Parameter estimation

Estimate the parameters $\Theta - \{\theta_1, \theta_2, ..., \theta_m\}$ that describes $f(X)$ probability density function of n i.i.d. random variables $x_1, x_2..., x_n$ by maximizing likelihood function $f(X|\Theta) = L(\Theta|X)$:

$$\hat{\Theta} = \underset{\theta \in \Theta}{argmax}\, \hat{L_n}(\Theta|x_i) = \underset{\theta \in \Theta}{argmax} \prod_i^n f_i(x_i|\Theta)$$

So equivalently applying the properties of logarithms to facilitate the calculation $\underset{\theta \in \Theta}{argmax} \sum_i^n ln(f_i(x_i|\Theta))$

1.1.1 1.1. Example

Estimate mean μ and std σ of 100 r.v. with $N(\mu, \sigma)$ distribution.

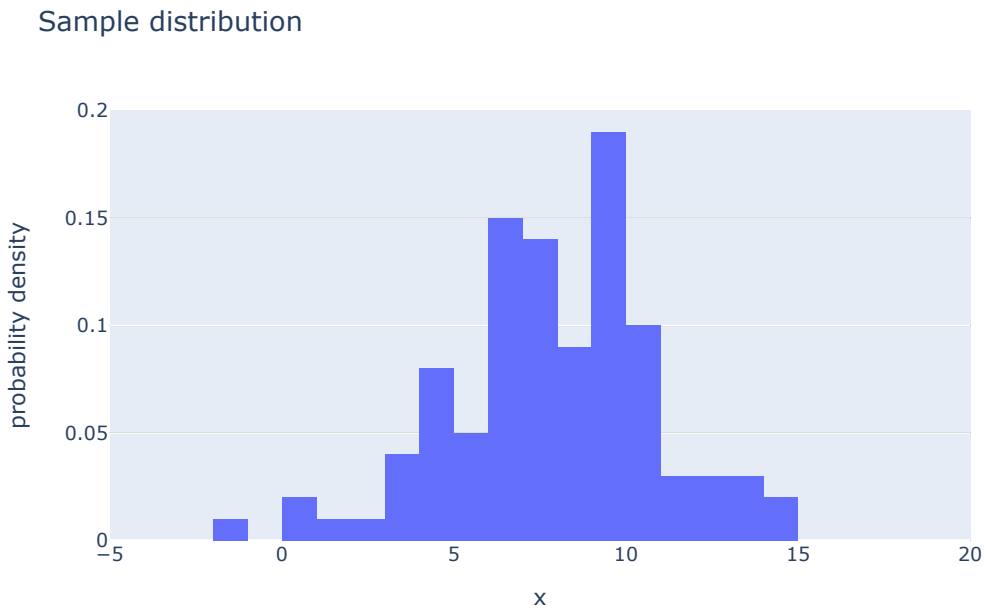
1º Create the sample vector $\mu = 8$ and $\sigma = 2.7$:

[10]:
```
samples = np.random.normal(8,2.7,100)
```

2º Representation of our samples:

[11]:
```
fig=px.histogram(x=samples,histnorm='probability density',title='Sample␣
↪distribution')
```

```
fig.update_xaxes(range=(-5,20))
fig.show()
```



3º Calculate likelihood in the parameter space $\sum_{i}^{n} ln(f_i(x_i|\mu,\sigma))$

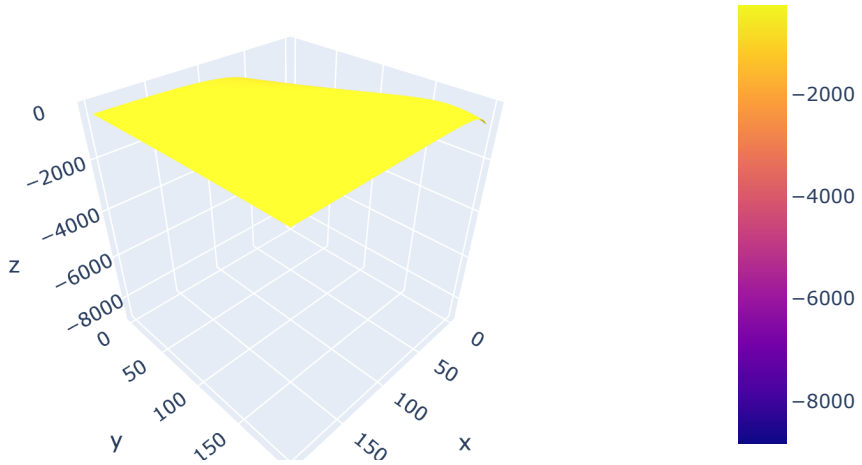
[12]:
```
means = np.linspace(-5, 10, 200, endpoint=True)
stds = np.linspace(1, 10, 200, endpoint=True)

l=np.empty([len(means), len(stds)])
for i in range(len(means)):
    for j in range(len(stds)):
        l[i,j]=(np.log(stats.norm.pdf(samples,loc=means[i],scale=stds[j])).
↪sum())

l[np.isnan(l)]=-1e-300
l[np.isinf(l)]=-1e-300
```

[13]:
```
fig = go.Figure(data=[go.Surface(z=l)])
fig.update_layout(title='MLE', autosize=False,
                  width=500, height=500,
                  margin=dict(l=65, r=50, b=65, t=90))
fig.show()
```

MLE



4º Extract the maximun $\underset{\theta \in \Theta}{argmax} \sum_{i}^{n} ln(f_i(x_i|\mu,\sigma))$

[14]:
```
row=np.argmax(np.max(l, axis=1))
col=np.argmax(np.max(l, axis=0))

print('Estimated mean: '+str(means[row]))
print('Estimated std: '+str(stds[col]))
```

```
Estimated mean: 7.889447236180903
Estimated std: 2.9899497487437188
```

1.2 1.2. Error

Calculation of the error in function of the number of samples

[15]:
```
mean=8
std=2.7

meanError=[]
stdError=[]

means = np.linspace(1, 10, 100, endpoint=True)
stds = np.linspace(1, 10, 100, endpoint=True)
```

```
for s in range(1,100):
    samples = np.random.normal(mean,std,s)

    l=np.empty([len(means), len(stds)])
    for i in range(len(means)):
        for j in range(len(stds)):
            l[i,j]=(np.log(stats.norm.pdf(samples,loc=means[i],scale=stds[j])).
 ↪sum())

    row=np.argmax(np.max(l, axis=1))
    col=np.argmax(np.max(l, axis=0))

    meanError.append(mean-means[row])
    stdError.append(std-stds[col])
```

[16]:
```
data0 = go.Scatter(x=np.linspace(0,len(meanError),len(meanError) ),␣
 ↪y=meanError, mode = 'lines', name='Abs. error of mean parameter')
data1 = go.Scatter(x=np.linspace(0,len(stdError),len(stdError) ), y=stdError,␣
 ↪mode = 'lines', name='Abs. error of std parameter')
data = [data0, data1]
layout = go.Layout(title='Parameter estimation error')
fig = go.Figure(data= data, layout = layout)
fig.update_xaxes(range=(0,100), title='Number of samples')
fig.update_yaxes(title='Abs. error')
fig.show()
```