HW1_Maximum_Likelihood_for_Gamma

CS249 — Spring 2016 — D.S. Parker

1 HW1: Maximum Likelihood Function Optimization

As in HW0, the basic problem here is to determine, given an input sequence of real values, which distribution it follows. More specifically, for this assignment you are to develop a program that reads in a numeric table, and – for each dataset (i.e., each column in the table) – determines the distribution and parameters that gives the closest match to it.

There are two differences between HW1 and HW0:

1. *in HW1, the input data are always drawn from the Gamma distribution.*
2. *in HW1, you must implement the Likelihood optimization yourself;*
   you cannot use `fitdistr()`*.*

As in HW0, your program could be given an input table like this:

D1	D2	D3	D4	D5	D6
3.3713903	6.2437282	0.2138276	0.1699299	1.5583491	0.6543210
2.7725880	5.5875745	0.4583172	0.3767378	2.8429449	1.9559299
⋮	⋮	⋮	⋮	⋮	⋮
7.2775941	5.2902876	0.9740191	2.6121070	5.9899608	6.7003783

The columns of this table define six datasets. Your program should produce a CSV file `HW1_output.csv` giving distributions that (it thinks) best fit the data. A correct output file could then look like this:

```
gamma,3,1
gamma,3,2
gamma,3,3
gamma,3,4
gamma,3,5
gamma,3,6
```

For simplicity, the parameters used in this assignment will always be integers, so the printed output should always have integer parameter values.

Your program can determine the distribution that fits best in any way you like. However, the notebook sketches a way to do this, and gives orientation about how to solve this problem in R.

In other words: yes, this is another simple assignment. It is intended as a warmup.

After running your program on the test input file `HW1_test.csv`, to complete this assignment please upload two files to CCLE:

1. your output CSV file `HW1_output.csv`
2. your notebook file `HW1_Fitting_Distributions.ipynb`

The notebook should have the commands you used to produce the output file. All assignment grading in this course will be automated, so please assume that when uploading files.

We will not execute your uploaded notebook. It should have the commands you used to produce the output file — in order to show your work. As announced, all assignment grading in this course will be automated, and the notebook is needed in order to check results of the grading program.

> Summary: the basic problem is to use Maximum Likelihood to determine, given a dataset of random real values, which distribution it follows. Your notebook should read in a numeric table, and − where each column in the table is a "dataset" − identify the distribution and Maximum Likelihood parameters that gives the closest match to it. Important Notes:
>
> - For simplicity, **the parameters in this assignment will always be integers.** Your printed output should always have integer parameter values.
> - **We will use Paul Eggert's Late Policy:** The number of days late is $N = 0$ for the first 24 hrs, $N = 1$ for the next 24 hrs, etc., and if you submit an assignment H hours late, $2^{\lfloor H/24 \rfloor}$ points are deducted.

1.1 Specific problem: Maximum Likelihood Parameter Fitting for the Gamma Distribution

A random variable x has some underlying probability distribution. The pdf $f(z)$ for this distribution usually depends on some parameters. If we call these parameters "θ" we can write the pdf as $f(z, \theta)$ or $f(z \mid \theta)$.

There can be multiple parameters, so θ can be a vector, and it is perfectly fine to have $\theta = (\theta_1, \theta_2, \theta_3)$ for example.

Here $f(z \mid \theta)$ reflects the probability of observing a given value z for the random variable x. Estimation is the process of finding values for the parameters θ, given some observations $x_1, \ldots, x_n$.

In Maximum Likelihood Estimation, the idea is to find the value of θ that maximizes the likelihood of the observations $x_1, \ldots, x_n$ having been observed.

Although $f(z \mid \theta)$ reflects the probability of observing a given value z for x, if we plug in the actually observed value x_i then $f(x_i \mid \theta)$ is not really a "probability". R.A. Fisher called $f(x_i \mid \theta)$ the likelihood of observing x_i. We want to find the value of θ that maximizes the likelihood function

$$likelihood(x_1, ..., x_n) \quad = \quad \prod_{i=1}^{n} f(x_i, \theta).$$

The value of θ that maximizes this function is called the maximum likelihood estimate (MLE).

1.2 Recall: Maximum Likelihood Fit for the Normal Distribution

Suppose f is a normal distribution with parameters $\theta = (\mu, \sigma)$, so that:

$$f(x, \theta) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right).$$

In this case the likelihood function is:

$$\text{likelihood}(x_1, ..., x_n) \quad = \quad \prod_i f(x_i, \theta) \quad = \quad \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp\left(-\frac{1}{2} \sum_i \left(\frac{x_i - \mu}{\sigma} \right)^2 \right).$$

The maximum likelihood estimate (value of θ that maximizes this function) clearly must minimize $\sum_i (x_i - \mu)^2$. By differentiating we can show that the minimum is obtained at

$$\mu \quad = \quad \frac{1}{n} \sum_i x_i.$$

In other words – as discussed in class – it turns out that the MLE is given by the usual formulas for μ and σ.

The maximum likelihood fit of a normal distribution to a dataset $x_1, \ldots, x_n$ has parameters that are the mean and variance of the data. Finding the best fit for a normal distribution reduces to this.

1.3 log-Likelihood

If the Likelihood is a product, its log (log-likelihood) is a sum. As a result it is easier to differentiate:

$$\log(\text{ likelihood}(x_1, ..., x_n) \ ) \quad = \quad \log(\ \prod_i f(x_i, \theta) \ ) \quad = \quad \sum_i \log \ f(x_i, \theta).$$

Since the logarithm function $\log(t)$ is a monotonic function of t (larger values of t yield larger values of $\log(t)$), by maximizing log(likelihood) we will also maximize (likelihood).

2 The Homework Problem: given an input set of data, find the MLE for the Gamma distribution

Given input values $x_1, \ldots, x_n$. we want to find the MLE parameters $\theta = (a, \beta)$, where α is a shape parameter andβ is a rate parameter.

The Gamma distribution has pdf

$$\frac{\beta^\alpha}{\Gamma(\alpha)} \ x^{\alpha-1} \ e^{-\beta x}$$

The distribution is described in the Wikipedia article on the Gamma Distribution.

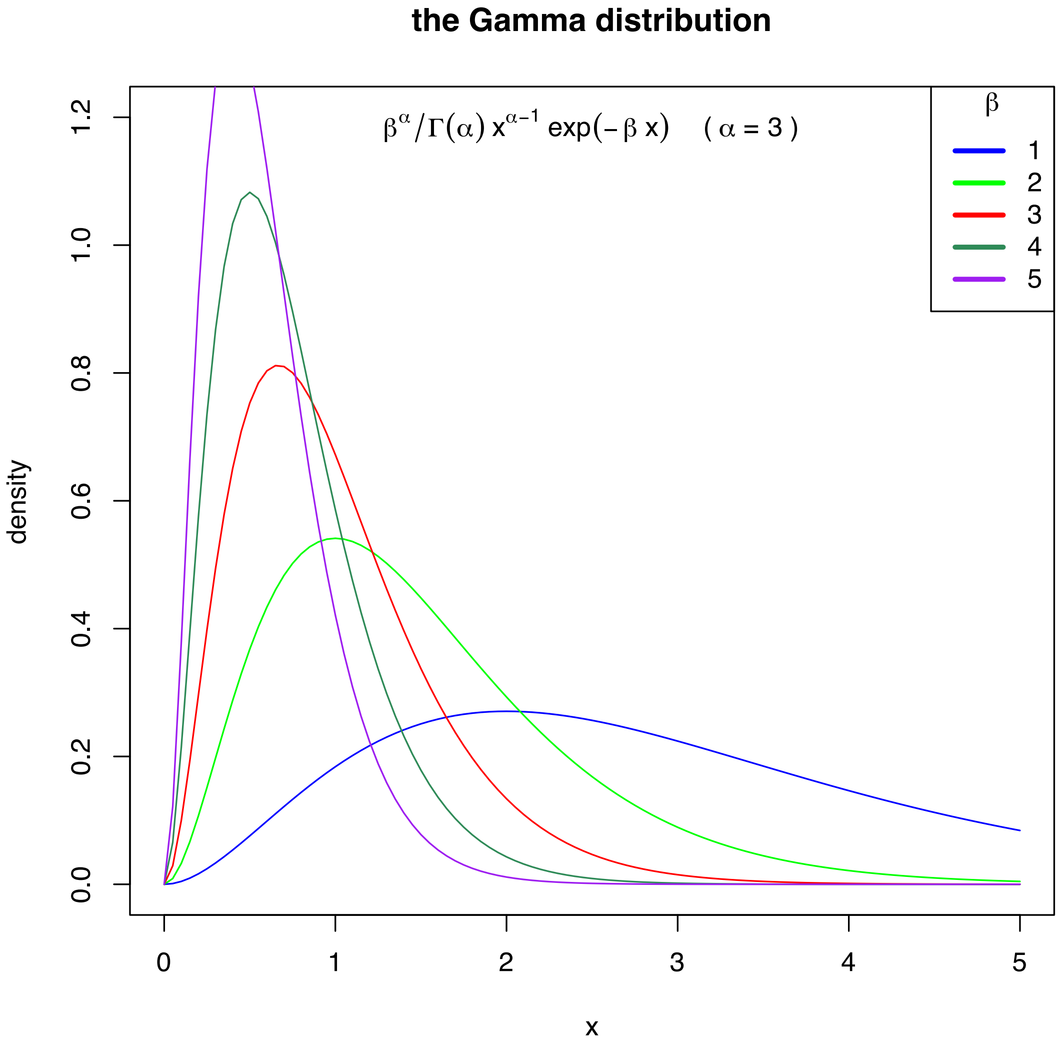
2.1 Plot some instances of the Gamma distribution

```
In [1]: plot( c(), c(), type="n", main="the Gamma distribution",
             xlab="x", ylab="density",
             xlim=c(0,5), ylim=c(0,1.2) )  # start the basic plot

        #  add a curve for each chi distribution parameter value:
        curve( dgamma(x, 3, rate=1), col="blue",     add=TRUE )
        curve( dgamma(x, 3, rate=2), col="green",    add=TRUE )
        curve( dgamma(x, 3, rate=3), col="red",      add=TRUE )
        curve( dgamma(x, 3, rate=4), col="seagreen", add=TRUE )
        curve( dgamma(x, 3, rate=5), col="purple",   add=TRUE )

        legend( "topright", paste(c(1,2,3,4,5)), title=expression(beta),
               col=c("blue","green","red","seagreen","purple"), lwd=3)

        mtext( expression( beta^alpha/Gamma(alpha) ~ x^{alpha-1} ~ exp( -beta ~ x )~
                          ~~~~ "(" ~ alpha ~ "=" ~ "3 )"),  side=3, line=-2)
```



3 The actual homework problem

You are given an $n \times p$ table as input. Each column of this table is a “dataset” of sample values x_1, …, x_n from a Gamma distribution. Your job is to find Maximum Likelihood Estimates for the parameters α, β for the Gamma distribution. For this problem you can assume that the parameters are always integers, so your program can round any non-integer value to the closest integer. Your program should obtain parameter estimates for each column in the table, and print the result as a sequence of lines; each line should have the format “gamma,α,β — where α, β are integers. For example, if we were given a table with values for the five values of β above, the output would be:

3.1 Example: the Log-Likelihood Function

```
In [2]: beta = 3
        alpha = 1
        mu = 0

        SampleDataset = rgamma(1000, alpha, rate=beta)
        # a sample dataset

        log_likelihood = function(theta) sum( log(dgamma(SampleDataset, theta[1], rate=theta[2])) )
```

```
In [3]: initial_value_for_theta = c(2.5, 2.8)
        negative_log_likelihood = function(theta) -log_likelihood(theta)

        # optim always _minimizes_ a function,
        #   so to find the MLE we minimize the negative log likelihood:

        # ? optim   # optimizer in R

        output_of_optimization = optim( initial_value_for_theta, negative_log_likelihood )

        print(output_of_optimization)
```

```
$par
[1] 0.9336271 2.8612689

$value
[1] -121.6437

$counts
function gradient
      97       NA

$convergence
[1] 0

$message
NULL
```

```
In [4]: ## minimum_negative_log_likelihood_value = output_of_optimization£value

        MLE_parameter_values = round( output_of_optimization$par )

        print(MLE_parameter_values)
```

```
[1] 1 3
```

```
In [5]: # Finally:  print out the estimate for nu in the format required:

        alpha = MLE_parameter_values[1]
        beta = MLE_parameter_values[2]

        cat(sprintf("Gamma distribution parameters: alpha = %d beta = %d\n", alpha, beta))
```

```
Gamma distribution parameters: alpha = 1 beta = 3
```

4 Problem: Fit a Gamma distribution to the input data.

Step 1. Extend the discussion above to fit a Gamma distribution to an input dataset

Your notebook should implement optimization of the Likelihood function.

Your R program might be an extension of this outline:

Step 2. The output of your program should be a CSV file "HW1_output.csv"

Your output CSV file "HW1_output.csv" should look like this:

If your program had been given the demo input file HW1_demo_input.csv as input, it should yield the following CSV file, a table with 6 rows, and THREE columns:

```
gamma,3,1
gamma,3,2
gamma,3,3
gamma,3,4
gamma,3,5
gamma,3,6
```

There should be NO header line in this file.

Each row has THREE fields: distribution name, and two parameter values.

If the input table has p columns (i.e., p random samples), the output file should have p rows.

The parameters in this assignment will always be integers, so the printed output should always have integer parameter values.

Step 3. Run your notebook using the file "HW1_test.csv" as input.

Step 4. Submit your output CSV file and notebook on CCLE.

Upload your .ipynb and your .csv file for Assignment "HW1". Both are required.

We will use Paul Eggert's Late Policy: The number of days late is $N = 0$ for the first 24 hrs, $N = 1$ for the next 24 hrs, etc., and if you submit an assignment H hours late, $2^{\lfloor H/24 \rfloor}$ points are deducted.