Lab/HW 7: Optimization

Your lab/homework must be submitted in with two files: (1) R Markdown format file; (2) a pdf or html file, unless otherwise stated. Other formats will not be accepted. Your responses must be supported by both textual explanations and the code you generate to produce your result.

Part I – Statistical Fitting Model: Method of Moments

The gamma distributions are a family of probability distributions defined by the density function,

$$f(x) = \frac{x^{a-1}e^{-x/s}}{s^a\Gamma(a)}$$

where the gamma function $\Gamma(a) = \int_0^\infty u^{a-1}e^{-u}du$ is chosen so that the total probability of all non-negative x is 1. The parameter a is called the ***shape*** and s is the ***scale***. When $a = 1$, this becomes the exponential distributions. The gamma probability density function is called `dgamma()` in R. You can prove (as a calculus exercise) that the expectation value of this distribution is as and the variance is as^2. If the mean and variance are known, μ and σ^2, then we can solve for the parameters,

$$a = \frac{a^2s^2}{as^2} = \frac{\mu^2}{\sigma^2},\ s = \frac{as^2}{as} = \frac{\sigma^2}{\mu}$$

In this lab, you will fit a gamma distribution to data using the **method of moments**. Our data today are measurements of the weight of the hearts of 144 cats.

1. [2 pt] The data is contained in a data frame called `cats`, in the R package `MASS`. This records the sex of each cat, its weight in kilograms, and the weight of its heart in grams. Run `summary(cats)` and explain the results.
2. [2 pt] Plot a histogram (normalized to a density) of the heart weights. Add a vertical line with your calculated mean (you may use `abline(v=yourmeanvaluehere)` for base graphics or `geom_vline(xintercept = yourmeanvaluehere)` if using `ggplot`.
3. [5 pt] Write a function that takes mean and variance as inputs and returns a and s, assuming that the mean and variance came from a random sample from a gamma distribution.
4. [5 pt] Calculate the mean, standard deviation, and variance of the heart weights using R's existing functions for these tasks. Use your function to get estimates of a and s. What are they?
5. [4 pt] Write a function, `cat_stats()`, which takes as input a vector of numbers and returns four estimates: mean, variance, a and s.
6. [4 pt] Estimate the a and s separately for all the male cats and all the female cats.
7. [4 pt] Now, produce a histogram for the female cats. On top of this, add the shape of the gamma pdf using `curve()` with its first argument as `dgamma()` (alternatively, use `ggplot` and `stat_function`). Is this distribution consistent with the empirical probability density of the histogram?
8. [4 pt] Repeat the previous step for male cats. How do the distributions compare?

Part II – Likelihood function

When we have independent samples $x_1, x_2, ...x_n$ from a common probability density $p(x)$, the joint probability density of the whole sample is $\prod_{i=1}^{n} p(x_i)$. When we are not sure what the right density p is, but we think it belongs to some family (like the Gaussian, the exponential, the gamma, etc.), we write the parameters of the family as θ, and say that the **likelihood function** is

$$L(\theta) = \prod_{i=1}^{n} p(x_i; \theta).$$

Notice that the likelihood is a **function of the unknown parameters** θ, not the known data $x_{1:n}$. One way to estimate the parameters is to maximize the likelihood,

$$\widehat{\theta}_{MLE} = \text{argmax}_\theta L(\theta).$$

For several reasons, including numerical stability, we usually work with the log-likelihood instead,

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{n} \log(p(x_i; \theta))$$

whose maximum is located at the same point as the maximum of L. By convention, optimization functions in software packages often find minimum values, so we often find maximum likelihood estimators by finding parameters to minimize the negative log-likelihood.

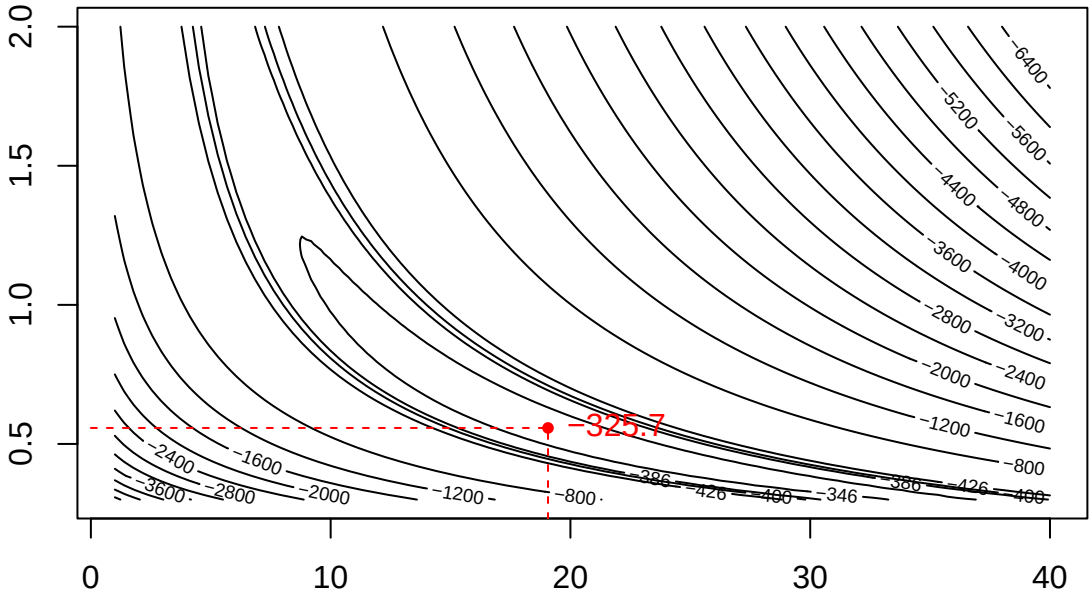
Maximum likelihood estimation is generally the most statistically efficient way to find the parameters of a probability density, when true density really is in the family we have guessed. In Part II, we begin working with likelihood functions, continuing to use the data on the heart weight of cats from the package *MASS*.

1. [5 pt] Calculate the log-likelihood of the shape and scale parameters you estimated, using the **method of moments** in Part I. The answer, rounded to the nearest integer, should be -326. Hint: `?dgamma`.
2. [8 pt] Write a function, `gamma_log_lik`, which takes in three inputs: `dat` - a numeric vector, a `shape` and a `scale` parameter (in that order), and returns the log-likelihood of the vector for the gamma distribution with the input shape and scale parameters. Check that when you run `gamma_loglike` with the heart weight data for cats and the estimates from Part I, you get the log-likelihood from question 1.
3. [10 pt] Using `curve()`, make two plots of the log-likelihood. In the first, vary the shape parameter over the range 1 to 40, while holding the scale parameter fixed at the value you estimated in Part I. In the second, vary the scale parameter over the range 0.2 to 2, holding shape fixed at your estimated value. At what values of the parameters do the two curves peak (you may use `which(vec == max(vec))` to find the index of the maximal value)? How do they compare to your estimates from Part I Q4? Are the two maxima equal? Should they be? Hint: remember that curve will sweep out over any argument, so long as you call it x and the function returns vector values.
4. [10 pt] Make a contour plot of the log-likelihood, with the shape parameter on the horizontal axis (range 1 to 40) and the scale parameter on the vertical (range 0.3 to 2). For this to work you need to create a new function, `gamma_cats` which admits only the shape and scale parameters, and applies `gamma_log_lik` to the heart weight data. In other words, `gamma_cats` is defined as a function that returns a function.

Add a point indicating the location of your moment-based estimate from question 1.

Also, you will probably want to increase the number of levels on the contour plot above the default of 10.

[3 pt Extra credit]: tweak your contour plot (use ?contour) so that the levels demonstrate the location of the maximum better, add a label with the value at the maximum log-likelihood (?text) and lines that indicate the levels of the maximizing parameters (?segments), as is demonstrated in the following:



[2 pt Extra credit] rewrite the contour function you choose to work with so that it utilizes `ggplot` with `geom_contour`

5. [3 pt] Is your contour plot from problem 4 compatible with your curves from problem 3? How can you tell?

6. [2 pt] Use the plot from the previous question to locate the region where the likelihood seems to be largest. Make a new plot which zooms in on this region by changing the ranges over which the shape and scale vary.

Part III – Maximize Likelihood

In Part III, we continue to use the data on the heart weight of cats from Part II.

1. [15 pt] Review Part II. Fit a gamma distribution to the cats' heart weights by maximum likelihood using `optim()`. Remember, the parameters for `optim` are in vector form (the `par` argument). You'll need to write a function that returns a function that accepts the parameters as a vector. Verify that the gradient of the negative log-likelihood at the maximum likelihood estimate is close to zero. Did you get better results compared to the method of moments estimates?

2. [15 pt] Perform the fit using the gradient descent function from class. Remember, the parameters are in vector form (the `theta0` argument). You'll need to write a function that returns a function that accepts the parameters as a vector. Did you get better results compared to the method of moments estimates? Compared to `optim`?

3. [5 pt Extra Credit] create a dense grid of shape and scale values around the estimates you found and see if you can further improve the result.