Exercises
Machine Learning
AS 2012

Machine Learning Laboratory
Dept. of Computer Science, ETH Zürich
Prof. Dr. Joachim M. Buhmann
Web http://ml2.inf.ethz.ch/courses/ml/

Series 2, Oct 9th, 2010 (Maximum Likelihood Estimation)

Email questions to: Gabriel Krummenacher
gabriel.krummenacher@inf.ethz.ch

Master solution will be available online from Tuesday, Oct 16th.

Problem 1 (Analytic MLE):

In this problem, we analytically derive maximum likelihood estimators for the parameters of an example model distribution. Most textbooks, including Duda et al, discuss the Gaussian example. The distribution we consider here is called the *gamma distribution*.

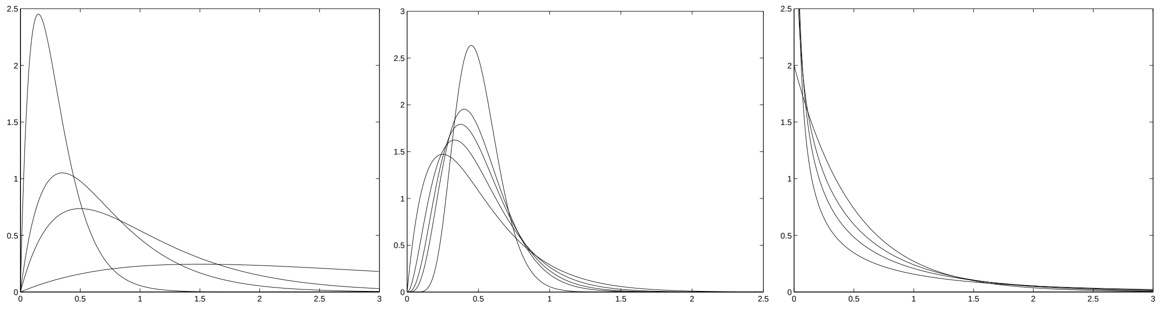
The gamma distribution is univariate (one-dimensional) and continuous. It is controlled by two parameters, the *location parameter* μ and the *shape parameter* ν.[1] For a gamma-distributed random variable X, we write $X \sim \mathcal{G}(\mu, \nu)$. $\mathcal{G}$ is defined by the following density function:

$$p(x|\mu,\nu) := \left(\frac{\nu}{\mu}\right)^{\nu} \frac{x^{\nu-1}}{\Gamma(\nu)} \exp\left(-\frac{\nu x}{\mu}\right) ,$$

where $x \geq 0$ and $\mu, \nu > 0$.[2] Whenever $\nu > 1$, the gamma density has a single peak, much like a Gaussian. Unlike the Gaussian, it is not symmetric. The first two moment statistics of the gamma distribution are given by

$$\mathsf{E}[X] = \mu \qquad \text{and} \qquad \mathsf{Var}[X] = \frac{\mu^2}{\nu} \tag{1}$$

for $X \sim \mathcal{G}(\mu, \nu)$. Here are some plots which should give you a rough idea of what the gamma density may look like and how different parameter values influence its behavior:



Left: The plot shows the density for different values of the location parameter ($\mu = 0.3, 0.5, 1.0, 3.0$), with the shape parameter fixed to $\nu = 2$. Since $\nu > 1$, the densities peak. As we increase μ, the peak moves to the right, and the curve flattens. *Middle:* For $\mu = 0.5$ fixed, we look at different values of the shape parameter ($\nu = 2, 3, 4, 5, 19$). Again, all the densities peak, with the peak shifting to the right as we increase ν. *Right:* If $\nu < 1$, the density turns into a monotonously decreasing function. The smaller the value of ν, the sharper the curve dips towards the origin.

1. Write the general analytic procedure to obtain the maximum likelihood estimator (including logarithmic transformation) in the form of a short algorithm or recipe. A few words are enough, but be precise: Write all important mathematical operations as formulae. Assume that data is given as an i. i. d. sample

[1] In parametric statistics, we usually call parameters shape parameters if they are neither location nor scale parameters.

[2] The symbol Γ denotes the distribution's namesake, the *gamma function*, defined by

$$\Gamma(\nu) := \int_0^\infty e^{-t} t^{\nu-1} dt .$$

The gamma function is a generalization of the factorial to the real line: $\Gamma(n) = (n-1)!$ for all $n \in \mathbf{N}$. Fortunately, we will not have to make explicit use of the integral.

$x_1, \ldots, x_n$. Denote the conditional density in question by $p(x|\theta)$, and the likelihood by $l(\theta)$. Make sure both symbols show up somewhere in your list, as well as a logarithm turning a product into a sum.

2. Derive the ML estimator for the location parameter μ, given data values $x_1, \ldots, x_n$. Conventionally, the ML estimator for a parameter is denoted by adding a hat symbol: $\hat{\mu}$. Given both the statistics of the gamma distribution (cf. (1)) and what you know about MLE for Gaussians, the result should not come as a surprise.

3. By now you should have some proficiency at deriving ML estimators, so a look at the gamma density will tell you that things get more complicated for the shape parameter: ν appears inside the gamma function, and both inside and outside the exponential. Thus, instead of deriving a formula of the form $\hat{\nu} := \ldots$, please show the following: Given an i. i. d. data sample $x_1, \ldots, x_n$ and the value of μ, the ML estimator $\hat{\nu}$ for the gamma distribution shape parameter solves the equation

$$\sum_{i=1}^{n} \left(\ln\left(\frac{x_i \hat{\nu}}{\mu}\right) - \left(\frac{x_i}{\mu} - 1\right) - \phi(\hat{\nu}) \right) = 0 . \tag{2}$$

The symbol ϕ is a shorthand notation for

$$\phi(\nu) := \frac{\frac{\partial \Gamma(\nu)}{\partial \nu}}{\Gamma(\nu)} . \tag{3}$$

In mathematics, ϕ is known as the *digamma function*.

Problem 2 (Numerical MLE):

This problem requires Matlab's optimization and statistics toolboxes, which should be part of the ETH installations.

One reason why ML estimation is appealing for use in everyday statistics is its generic applicability. ML estimators may be derived analytically for theoretical considerations, but to simply fit a model, one may as well rely on numerical methods: Suppose we consider ML estimation an appropriate approach for a given problem, and that we know the functional form of the likelihood. Then all we have to do in practice is to implement this likelihood in Matlab and apply the numerical optimization functions from the optimization toolbox.

We ask you to implement numerical ML estimation for the parameters of the gamma distribution, because there exists no closed form expression for the shape parameter:

1. Implement the log-likelihood of the gamma distribution as Matlab function `llhood_gamma(x, theta)`. $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ contains the samples and $\theta = (\mu, \nu)$ contains the location and shape parameter. Use the function `gamma` to compute $\Gamma(\nu)$.

2. Implement a second function `mlest(llfun, x)` which takes two inputs:

   - `llfun`: A function handle to any log-likelihood $l(\mathbf{x}, \theta) := \sum_i \log p(x_i; \theta)$ which obeys the generic signature `llfun(x, theta)`. `llhood_gamma` from above is one such function, but by using a function handle instead of a hard-coded call, you can use `mlest` for any log-likelihood function.
   - `x` : the vector of data samples.

   Note that the sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ must be handed to `mlest` as an argument, in order that it can evaluate `llfun` on `x`. If you are unfamiliar on how to use function handles, i.e. passing references to functions as arguments, have a look at the following Matlab help topics: `function_handle, fminsearch`.

   Calling optimization procedures in Matlab takes some getting used to. Please take the following technical issues into account:

   - Since the optimization toolbox provides only minimization procedures, you will have to transform the maximization problem into a minimization problem. Don't forget to reversely transform the result.
   - The likelihood function `llfun` takes two parameters: the sample $\mathbf{x}$ and the model paramters θ, but the maximization procedure targets only θ. Therefore, we use the so-called anonymous function handle: the calling argument of the optimization routine `fminsearch` should be of the form `(@(theta) llfun(x, theta))`. This tells the optimization to specifically target the parameter `theta`. Again, have a look at the help description of the optimization routine, which provides examples for this type of function call.

- `fminsearch` takes as its second parameter an initial guess θ_0, from where it starts the optimization process. Depending on θ_0, you may have to increase the maximum number of iterations and function evaluations that `fminsearch` performs to get a good result. The documentation of `optimset` shows you how to do it.

3. To test your implementation, produce a 1D gamma sample using `gamrnd(a, b, n, d)` from the statistics toolbox. It takes 4 inputs where $a = \nu$, $b = \frac{\mu}{\nu}$, and (n, d) are the dimensions of the output data. In our case, set $d = 1$.

   Pass both the handle to the gamma log-likelihood `llhood_gamma` and the sample to your `mlest` implementation. Compare your numerical estimates with true values (used in `gamrnd`) and to `mean(x)`, which coincides with the analytical solution for $\hat{\mu}$, and `var(x)`, which should be close to $\hat{\mu}^2/\hat{\nu}$. Try different parameter values and sample sizes. How close to the true value does your initial guess θ_0 have to be?

Here is what we ask you to write down:

- Code for `llhood_gamma` and `mlest`
- Your results for Question 2.3 and the conclusions you draw from them

Problem 3 (Terminology problems):

As consequence of the interdisciplinary character of machine learning, terminology is not always used consistently throughout the machine learning community. We will consider an example of the term "maximum likelihood estimation" being applied in a manner which is, from a statistician's point of view, not quite correct.

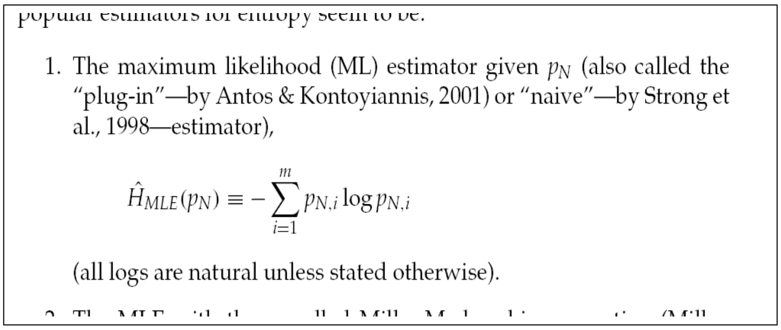
A statistic of some importance is the *entropy* H, defined for a discrete random variable X with mass function P and sample space $\mathcal{X}$ as

$$H(X) := -\sum_{x \in \mathcal{X}} P(x) \log(P(x)) \ . \tag{4}$$

Over the last few years, several publications have considered how to reliably estimate this statistic from a sample. The following is a definition taken from one of these publications. Only two assumptions are being made:

- An arbitray data sample of size N is given.
- The symbol p_N denotes the *empirical* distribution of the sample in question (with $p_{N,i}$ the empirical probability at the ith sampling point).

Here is the definition from the article:



popular estimators for entropy seem to be:

1. The maximum likelihood (ML) estimator given p_N (also called the "plug-in"—by Antos & Kontoyiannis, 2001) or "naive"—by Strong et al., 1998—estimator),

$$\hat{H}_{MLE}(p_N) \equiv -\sum_{i=1}^{m} p_{N,i} \log p_{N,i}$$

(all logs are natural unless stated otherwise).

2. The MLE with the so-called Miller–Madow bias correction (Mill

1. From the point of view of statistics, the cited works by Antos & Kontoyiannis and Strong et al were right *not* to call this a maximum likelihood estimator. Can you explain why the term "maximum likelihood estimator" is not properly used here?
2. Which further assumptions regarding the underlying distribution would we have to make in order to correctly define a maximum likelihood estimator for the entropy (or any other statistic)?