ST4060 Project

Cian Scannell
University College Cork

On the implementation and testing of a semi-parametric minimum entropy estimator for regression analysis

July 30, 2016

Abstract

The aim of this project was to identify, implement and test a semi-parametric entropy estimator for the purpose of regression analysis. This was primarily achieved via study of research papers authored by E. Wolsztynski, E. Thierry and L. Pronzato [1,3], and subsequent replication of various examples outlined in [1].

Introduction

We consider the nonlinear regression model

$$y_i = \eta(\bar{\boldsymbol{\theta}}, \xi_i) + \epsilon_i, \ i = 1, ..., n,$$

where the y_i are the observations, $\bar{\boldsymbol{\theta}}$ is the unknown true value of the model parameters $\boldsymbol{\theta}$, $\eta(\boldsymbol{\theta}, \xi_i)$ is a known function of $\boldsymbol{\theta}$ and the design variable ξ and the ϵ_i are i.i.d random variables with pdf f. Note that the residuals are defined as

$$e_i(\boldsymbol{\theta}) = y_i - \eta(\boldsymbol{\theta}, \xi_i)$$

Our goal is to estimate $\boldsymbol{\theta}$ by minimising the dispersion of the residuals $e_1(\boldsymbol{\theta}), ...e_n(\boldsymbol{\theta})$ (denoted $\boldsymbol{e}_1^n(\boldsymbol{\theta})$). The case where f is known has been widely studied, often using methods we have discussed in class, such as least squares and maximum likelihood estimation [2]. We know that the maximum likelihood estimator $\hat{\boldsymbol{\theta}}_{ML}$ minimises

$$\bar{H}_n(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n} \log f(e_i(\boldsymbol{\theta}))$$

Which is essentially a discretised version of Shannon Entropy,

$$ent(f) = -\int_{-\infty}^{\infty} f(x) \log f(x) dx$$

A natural extension to this idea is to base our estimation criterion on the idea of entropy. It also allows us to deal with the important case where f is unknown. In addition to the stark mathematical relationship, there is also strong intuition behind the use of an entropy based criterion. After all, entropy can simply be thought of as a measure of dispersion. Minimising the dispersion of the residuals will force them to gather. To clarify, our goal is now to estimate $\boldsymbol{\theta}$ by minimising the entropy of an estimate of f, based on the empirical distribution of the residuals $\boldsymbol{e}_1^n(\boldsymbol{\theta})$. (This is why our estimator is termed semi-parametric, as it is comprised of both a parametric ($\boldsymbol{\theta}$) and a non-parametric (f) element.)

Method

In our quest to estimate f we use our wealth of knowledge accumulated during the course of ST4060. We introduce a kernel density estimate based on residuals.

$$\hat{f}_{n,h}(x) = \hat{f}_{n,h}(x \mid e_1(\boldsymbol{\theta}), ..., e_n(\boldsymbol{\theta}))$$

$$= \hat{f}_{n,h}(x \mid \boldsymbol{e}_1^n(\boldsymbol{\theta}))$$

$$= \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - e_i(\boldsymbol{\theta})}{h}),$$

where h is the bandwidth and K is a kernel function which possesses the usual properties, i.e. K is non-negative, integrates to 1 and has a mean of zero.

There is one problem left to tackle before we begin the implementation - that of the invariance of $ent(f)$ under translation. We shall omit full details of this as it beyond the scope of this course. Suffice to note, however, that minimising

$$J_e(\boldsymbol{\theta}) = ent(\hat{f}_{n,h}(\ .\ |\ \boldsymbol{e}_1^n(\boldsymbol{\theta})))$$

with $\boldsymbol{\theta} = (\theta_1, ...\theta_p)$, will make it impossible to recover θ_1 in certain instances. So we use

$$J_e^s(\boldsymbol{\theta}) = ent(\hat{f}_{n,h}(\ .\ |\ \boldsymbol{e}_1^n(\boldsymbol{\theta}), -\boldsymbol{e}_1^n(\boldsymbol{\theta})))$$

Implementation

The above method was implemented using R (v. 3.2.2).

We defined functions to calculate the residuals and the kernel. The criterion to be minimised was estimated using a kernel estimate based on the symmetrised residuals [3].

$$J_e^s(\theta) = -\frac{1}{N}\sum_{j=1}^{N} \log(\frac{1}{2Nh}\sum_{i=1}^{N}[K(\frac{e_j(\theta) - e_i(\theta)}{h}) + K(\frac{e_j(\theta) + e_i(\theta)}{h})])$$

This was implemented by segmentation into smaller functions, as can been seen in our R code.
Various other functions were created for testing purposes. Methods such as Least Squares estimation and Inverse transform were implemented.

Testing

Example 1. We first considered the nonlinear regression model with $\eta(\theta, \xi) = \exp(-\theta\xi)$ [1]. We took 20 observations at $\xi = 0.5, 0.55, \ldots, 1.45$ and assumed that our kernel was given by

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\,(\frac{-u^2}{2})$$

With the assumption that the errors ϵ_i are i.i.d $\mathcal{N}(0, \sigma^2 = 0.01)$ we were able to implement a Monte Carlo simulation where we obtained both the Maximum Likelihood estimator $\hat{\theta}_{ML}$ (which is equivalent to the Least Squares estimator in this situation) and the Entropy estimator $\hat{\theta}_e$ with h fixed to be 0.1. *Figure 1* shows a very good agreement between the two.

Example 2. To continue with our investigation we then considered the same model with the errors ϵ_i now independently and uniformly distributed on $[-0.4, 0, 4]$. The entropy estimator seems to be significantly more accurate than the LS-estimator. *Figure 2* shows a

Figure 1. Empirical densities of $\hat{\theta}_{ML}$ and $\hat{\theta}_e$ with $h = 0.1$ for 1000 repetitions, assuming the errors are normally distributed (Example 1).

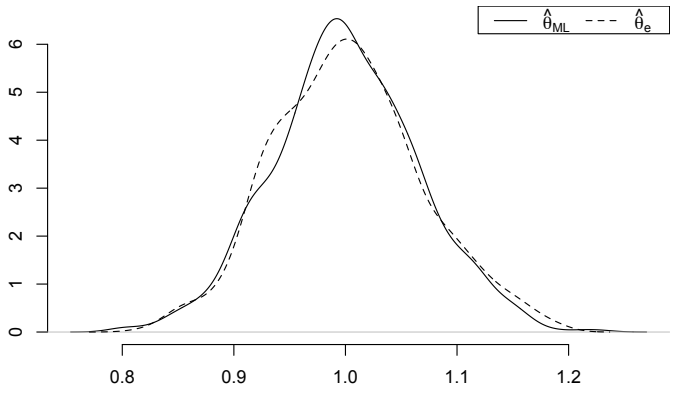


Figure 2. Empirical densities of $\hat{\theta}_{LS}$ and $\hat{\theta}_e$ with $h = 0.1$ for 1000 repetitions, assuming the errors are uniformly distributed (Example 2)

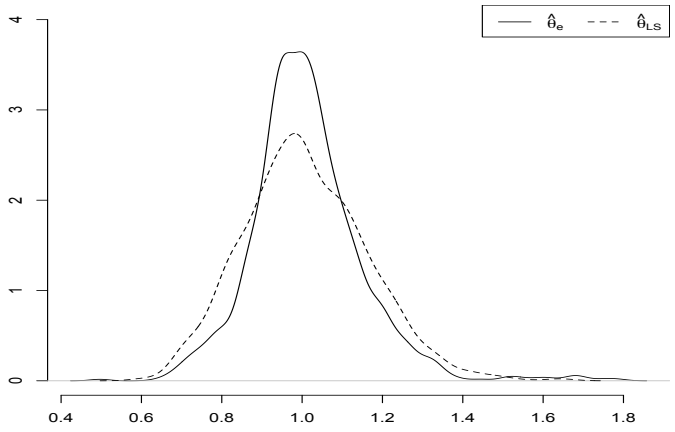
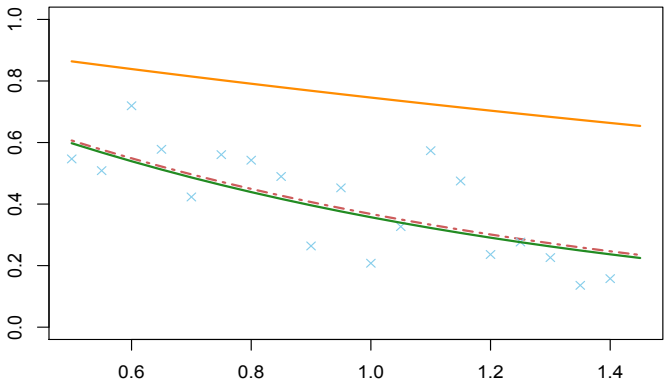


Figure 3. The model for different values of θ. The dashed line is where we use $\bar{\theta}$, the orange (top) is for $\hat{\theta}_{LS}$ and the green is for $\hat{\theta}_e$ with $h = 0.1$. (Example 4)



clear difference between the two empirical densities with $\hat{\theta}_{LS}$ having the larger variance.

Example 3. We then considered the case when the errors were i.i.d with a Laplace density $f(x) = 1/(\sigma\sqrt{2})\exp\big(-\sqrt{2}|x|/\sigma\big)$. We employed the inverse transform method to allow us generate a random sample from this distribution using the following scheme.

- Generate u and u' a random sample from the uniform distribution on $(0, 1)$
- If $u'[i] < 0.5$ return $\frac{\sigma}{\sqrt{2}}\log(2u)$
- Else return $-\frac{\sigma}{\sqrt{2}}\log(2u)$

Again for 1000 repetitions we saw that the variance associated with our simulations of the Entropy estimator was rough 0.00304 and for the LS-estimator this increased to 0.00493. This once more verifies the superior accuracy of the Entropy estimator.

Another property of the estimator we would like to test is its robustness. In particular we would like to ascertain how the estimator performs under changes. We implemented a further example where we include an outlier and again compare the two estimators.

Example 4. Here we consider the exact same situation as example 3 except we replaced the last $y-$value with an outlier, $y_{20} = 5$. *Figure 3* displays the results of this testing. We plotted the curves for the model we get for both $\hat{\theta}_e$ and $\hat{\theta}_{LS}$. We see that we still maintain an excellent fit for $\hat{\theta}_e$, much better than for $\hat{\theta}_{LS}$ which is strongly effected by the introduction of the outlier.

Example 5. In this example, we consider the necessity of symmetrising the entropy estimator criterion. Using the data set *'Tombstone Weathering'*[4] we fitted a linear

regression model,

$$y_i = \theta_0 + \theta_1 x_i$$

Our criterion function used previously was adjusted to create an unsymmetrised version. Using the optim function in R, the parameters of the model were calculated using the symmetrised criterion, the unsymmetrised criterion and the least squares method (this acted as our control mechanism). The calculated values are recorded in table 1.

Table 1
Parameter estimates for the various methods used.

Method	θ_0	θ_1
Least Squares	0.32331	0.00859
Unsymmetrised	0.19383	0.00865
Symmetrised	0.29231	0.00861

Since our model is in the form $\eta(\boldsymbol{\theta}, \xi) = \eta_1(\theta_0) + \eta_2(\theta_1, \xi)$ we can not estimate θ_0 using the unsymmetrised version [1]. Indeed, we can see that, although parameter estimates for θ_1 are reasonably similar across all methods, there is a large discrepancy for estimates of θ_0 between the least squares and unsymmetrised criterion methods (difference of 0.129), while the θ_0 estimates for the least squares and symmetrised criterion methods are roughly similar (difference of 0.031).

We note however that these results we find are sensitive to our initial starting point. Perhaps there is a problem with the convexity of our function in this case and our optimization techniques only find a local optimum. Another possibility is the unsuitability of the Gaussian kernel applied to this dataset. Or else our implementation leaves something to be desired.

References

[1] *A minimum-entropy estimator for regression problems with unknown distribution of observation errors*, L. Pronzato & E. Thierry, 2000.

[2] *Lecture Notes for ST4060 taught at University College Cork*, E. Wolsztynski, 2015.

[3] *Minimum-Entropy Estimation in Semi-Parametric Models*, E. Wolsztynski, E. Thierry & L. Pronzato, 2004.

[4] `www.stat.ufl.edu/~winner/datasets.html`