M2TSI - SpaceMaster

UE31 Laboratory Report

Signal Estimation

Authors:
Arthur Scharf
Andreas Wenzel

January 10, 2017

1 Introduction

In this report various methods for the estimation of parameters of a given data set are evaluated and compared. To cut down the problem's complexity, we use a simple model for the flux of an elliptic galaxy, as is described by the Sersic profile, which provides us an initial data set. This data set - a noisy image of a elliptic galaxy as it would have been taken by a ground-based telescope - is then used to evaluate different estimation approaches as the least square estimation, maximum likelihood estimation and Bayesian estimation.

2 Modelling a galaxy

The Sersic profile is very common amongst astrophysicists to model the flux of observed elliptic galaxies in a simple way, and is given by the equation

$$I(l,c) = exp(-R(l,c)^{\frac{1}{n}}) \tag{1}$$

which describes the variation of intensity with respect to the distance of the galaxy's centre. The distance R of a pixel with the coordinates (l,c) from the galaxies centre is given by

$$R(l,c)^2 = \left(\frac{(l-l_0)\sin(\alpha) - (c-c_0)\cos(\alpha)}{\sigma_l}\right)^2 + \left(\frac{(l-l_0)\cos(\alpha) - (c-c_0)\sin(\alpha)}{\sigma_c}\right)^2 \tag{2}$$

with (l_0, c_0) being the galaxy's centre coordinates, (σ_l, σ_c) the two galaxy's axes length and the horizontal angle α.

This leads to the following equation modelling the data

$$d(l,c) = s + aI(l,c) + n(l,c) \tag{3}$$

with a as the amplitude of the galaxy, s the amplitude of the sky's background and $n(l,c)$ as noise.

Using the Sersic profile we can now generate an artificial elliptic galaxy, see fig. 1.

By assuming a Gaussian white noise with the known variance σ_n^2 and using the initial image of the galaxy, we can model the elliptic galaxy as it would have been seen by a ground-based telescope, see fig. 2.

The parameters used to create above mentioned figure are given as follows

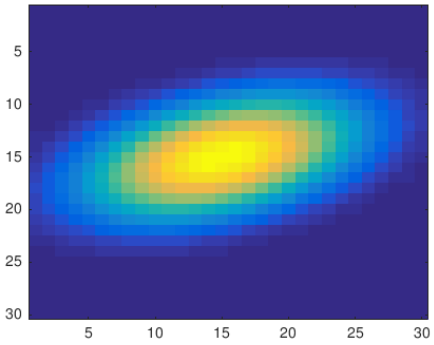

Figure 1: Initial data set

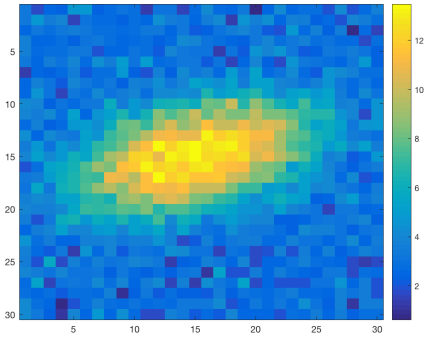

Figure 2: Random noise added

$$L, C = 30 \qquad l_0, c_0 = 15 \qquad \sigma_l = 10 \qquad \sigma_c = 5 \qquad \sigma_n = 0.8$$
$$\alpha = 0.3 \qquad n = 0.5 \qquad a = 10 \qquad s = 3$$

with L, C as height respectively width of image in pixels and σ_n as the standard deviation of the gaussian white noise.

3 Estimation with known Galaxy's location and shape parameters

Having successfully modelled a simple elliptic galaxy, this model is used as input data for the simulation of an estimation of the galaxy's and sky background's amplitude, while assuming that the galaxy's position and shape parameters are known. This estimation can be formulated via:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{n}(c, l) \tag{4}$$

Here, $\boldsymbol{y}$ is the actual measurement, in our case it is the intensity of the galaxy. However, one has to keep in mind that for the following steps, we use the simulated galaxy intensity as described above as our measured galaxy intensity. $\boldsymbol{\theta}$ is the vector of estimated parameters, which are in our case the galaxy's amplitude a and the background's amplitude s, so $\boldsymbol{\theta} = [s, a]^{\mathrm{T}}$. $\boldsymbol{H}$ is a matrix describing the underlying model, which in our case also depends on the pixel position since the intensity described by the sersic function depends on the distance towards the centre of the galaxy, and $\boldsymbol{n}(l, c)$ is a noise component also depending on the pixel position (l,c). In order to satisfy equation 3 using equation 6, one has to define $\boldsymbol{H}$ in a special matrix form, which is given by:

$$H_{l,c} = \begin{pmatrix} 1 & \exp(-R_{l,c}^{\frac{1}{n}}) \end{pmatrix} \tag{5}$$

Thus, $\boldsymbol{H}$ is a $2 \times (L \cdot C)$ matrix, describing the intensity profile of the galaxy for each pixel (l,c). Therefore, our data $\boldsymbol{d}$ describing the measured data for all pixels yields:

$$\boldsymbol{d} = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{n}(c, l), \tag{6}$$

where d is a vector with $L \cdot C$ elements.

In order the obtain maximum likelihood estimator for $\boldsymbol{\theta}$ as a function of $\boldsymbol{d}$, one has to calculate:

$$\boldsymbol{\theta} = (\boldsymbol{H'H})^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{d} \tag{7}$$

The bias of that estimator could be calculated using $b_{\boldsymbol{\theta}}(\boldsymbol{\theta}_{init}) = E_{Y|\boldsymbol{\theta}_{init}}(\boldsymbol{\theta}) - \boldsymbol{\theta}_{init}$, where $\boldsymbol{\theta}_{init}$ is the vector of the true parameters that we want to find out. However, this bias is only applicable for a larger sets of simulations. The used matlab code performing this is shown in the appendix. The resulting maximum likelihood estimators for the galaxy's and the sky background's amplitudes are:

$$\boldsymbol{\theta} = [2.9576\ 10.0131]^{\mathrm{T}}$$

This result is very close to the original parameters used for the simulation of the data with $\boldsymbol{\theta}_{initial} = [3\ 10]^{\mathrm{T}}$ (see list of parameters above). However, we assumed the galaxy center and shape to be known, which drastically reduces the number of unknowns in this fit, and therefore results in a very good fit result.

4 Maximum likelihood estimation of all parameters

In the next step, we assumed that the galaxy's position and shape are unknown for the fit. Thus, now also the parameter vector $\boldsymbol{\nu} = [l_0, c_0, \sigma_l, \sigma_c, n]$ also needs to be fitted simultaniously to θ. Again, the corresponding model can be formulated as:

$$\boldsymbol{d} = \boldsymbol{H}(\boldsymbol{\nu})\boldsymbol{\theta} + \boldsymbol{n}(c, l),$$

The only difference now is that $\boldsymbol{H}$ depends on the variable $\boldsymbol{\nu}$. Again, $\boldsymbol{H}$ is a $2 \times (Ł \cdot C)$ - matrix, $\theta = [s, a]^{\mathrm{T}}$, $\boldsymbol{d}$ a vector with $L \cdot C$ components and $\$\boldsymbol{\nu}$ as defined above a vector with six components. The goal is to obtain the maximum likelihood estimator by minimizing the quadratic cost function $\boldsymbol{J}$:

$$\boldsymbol{J}(\boldsymbol{\theta}, \boldsymbol{\nu}) = (\boldsymbol{d} - \boldsymbol{H}(\boldsymbol{\nu})\boldsymbol{\theta})^{\mathrm{T}} (\boldsymbol{d} - \boldsymbol{H}(\boldsymbol{\nu})\boldsymbol{\theta}) \tag{8}$$

The optimization problem thus is a least square optimization. The maximum likelihood estimators are then given by:

$$(\boldsymbol{\theta}_{ML}, \boldsymbol{\nu}_{ML}) = argmin\,(\boldsymbol{J})$$

For a large number of samples, this maximum likelihood estimator is unbiased with a variance equal to the inverse of the Fisher matrix: $\boldsymbol{\sigma}^2 = \boldsymbol{F}(\boldsymbol{\theta})^{-1}$

The simplest method to gain initial parameters for the optimisation is to perform an "educated guess" using the obtained noisy image from the galaxy. More specifically, the galaxy's centre and the amplitude of the galaxy as well as the background noise can be read straight from the image within a reasonable error-range. However, the shape and form of the galaxy can not be read out or estimated directly since the present noise can not be estimated with a reasonable accuracy. Thus we use arbitrary values for the shape parameters.
These initial parameters can then be used to compute a cost function, c.f. appendix C, which then can be minimised to obtain an optimal set of parameters based on the initial ones. Re-inserting the former into the Sersic model, we can generate an estimation of the true galaxy (c.f. fig. 1), as can be seen in fig. 3.

	s	a	l_0	c_0	σ_l	σ_c	α	n
initial	1	20	14	17	1	1	1	1
estimated	3.0660	9.8379	14.9496	15.0309	10.0970	5.0054	0.2978	0.4922
theoretical	3	10	15	15	10	5	0.3	0.5

Table 1: parameter comparison

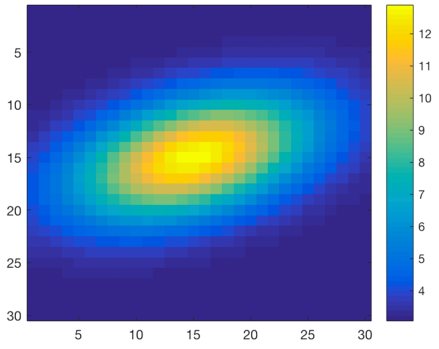

Figure 3: estimated galaxy

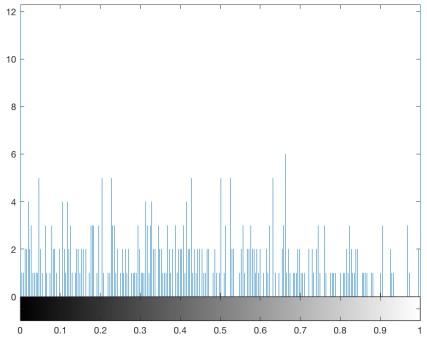

Figure 4: Histogram of residual data

As can be seen in table 1, the estimated parameters are extremely close to the theoretical ones that were used to obtain fig. 1, our initial image of the elliptic galaxy. This results in an image (c.f. fig. 3) of the galaxy that is very similar to the initial one, even though the initially guessed parameters were not chosen close to the theoretical ones.

Since this optimisation approach is very susceptible to the initial parameter guess, it is not guaranteed that this approach will succeed and result in physically reasonable optimal parameters. To avoid this behaviour in practice, the initial parameters can be altered until they are within an acceptable range or the optimisation process can be modified such that physically unreasonable parameters are rejected during the process - which adversely would increase the complexity of the whole process.

By subtracting the image created with the estimated parameters from the noisy image presumably obtained by the telescope (fig. 2), we obtain the so-called "residuals", which correspond to

the noise that is included in the obtained image. Considering the histogram of these residuals as shown in fig. 4 and by calculating the standard deviation (c.f. Matlab code in appendix B), the noise seems to be compatible with our model for the noise - a simple white gaussian noise - since the histogram shows a more or less even distribution of numerical data and the standard deviation of 0.7949 is very close to our initially introduced value of $\sigma = 0.8$.

However, varying the initial parameters for the optimisation can result in unusable data - which can not even be interpreted as elliptic galaxy anymore, especially if the initial parameters are too different from the theoretical ones. The closer the parameters are chosen to their respective theoretical value the better the maximum likelihood estimation performs.

A Sersic function

```
function Im = Sersic(nu,Lig,Col)

% Im = Sersic(nu,Lig,Col)
%
% Calcul de l'image d'un module de Sersic sur une grille (definie par les
% matrices Lig et Col) de parametres contenus dans le vecteur nu:
%  - nu(1) = l_0 : position en ligne
%  - nu(2) = c_0 : position en colonne
%  - nu(3) = sigma_l : cart-type en ligne
%  - nu(4) = sigma_c : cart-type en colonnne
%  - nu(5) = angle : angle en radian
%  - nu(6) =  n : indice de Sersic

% H. Carfantan, IRAP, novembre 2014

l_0 = nu(1);
c_0 = nu(2);
sigma_l=nu(3);
sigma_c=nu(4);
n = nu(6);
alpha = nu(5);
Im = exp(-  (  (((Col-c_0)*cos(alpha)-(Lig-l_0)*sin(alpha))/sigma_l).^2 ...
             + (((Col-c_0)*sin(alpha)+(Lig-l_0)*cos(alpha))/sigma_c).^2).^(1/(2*n)) );
```

B Parameter Estimation

```
clear all, clc, close all;

% Modeling the problem
L = 30;
C = 30;
l0 = floor(L/2);
c0 = floor(C/2);
sigma_l = 10;
sigma_c = 5;
alpha = 0.3;
n = 0.5;
```

```
12 a = 10;
13 s = 3;
14
15 % noise parameter
16 sigma = 0.8;
17
18 % creating the galaxy
19 lin = 1:L;
20 col = 1:C;
21 [Col,Lin] = meshgrid(col,lin);
22 nu = [l0; c0; sigma_l; sigma_c; alpha; n];
23 Gal = Sersic(nu,Lin,Col);
24
25 % adding random noise to the initial Galaxy Image
26 D = s + a*Gal + (sigma*randn(L,C));
27 d = D(:);
28 D = reshape(d,L,C);
29
30 % plot the initial data sets
31 figure
32 subplot(2,2,1), imagesc(Gal), title('initial data set')
33 subplot(2,2,2), imagesc(D), title('initial set with added noise')
34 colorbar
35
36 % now we write matrix H
37 H = [ones(L*C,1), Gal(:)];
38 ML = inv(transpose(H)*H)*transpose(H)*d
39
40 % question 3, has not to be done here - theoretical question
41 real = [s a].'
42 bias = ML - real
43
44 % we create our p-vector that contains theta and nu parameters
45 % p_init = [s; a; nu];
46 p_init = [1;20;14;17;1;1;1;0.5];
47
48 % start optimisation using the crit_J cost function
49 options = optimset('fminsearch');
50 p_opt = fminsearch(@(p) crit_J(p,D), p_init,options)
51 Gal = Sersic(p_opt(3:end),Lin,Col);
52 D_opt = p_opt(1) + p_opt(2)*Gal ;
53
```

```
54 d = D_opt(:);
55 D_opt = reshape(d,L,C);
56
57 subplot(2,2,3), imagesc(D_opt), title('estimated data set')
58 colorbar
59
60 % calculate image residuals from estimated parameters and show
      histogram
61 Res = D - D_opt;
62 stdRes = std2(Res)
63 subplot(2,2,4)
64 imhist(Res)
65 ylim([0 max(Res(:))+10])
66 title(sprintf('Residual histogram, stdDev: %0.3f',stdRes))
```

C crit_J cost function

```
1 % crit_J function to calculate cost function (to minimize it)
2 function J = crit_J(p, D)
3     nu = p(3:end);
4     theta = transpose([p(1) p(2)]);
5     [m n] = size(D);
6     [Col,Lin] = meshgrid(1:m,1:n);
7     Gal = Sersic(nu,Lin,Col);
8
9     H = [ones(m*n,1), Gal(:)];
10    J = transpose(D(:) - H*theta)*(D(:) - H*theta);
11 end
```