Loss Functions and Robustness in Data Fitting

Generalized Nonlinear Data Fitting Solver from the NAG Library

Data Fitting Introduction

Fitting a non-linear model to data is typically modelled as a minimisation problem, where the objective function serves as a measurement of the quality of the model's fit to data, depending on our parameters. A general model involves summing over our data points,

$$\underset{x \in \mathbb{R}^{n_{\text{var}}}}{\text{minimize}}\ f(x) = \sum_{i=1}^{n_{\text{res}}} \chi(r_i(x)),$$

where x is a vector holding our model parameters, of which there are n_{var}. We have n_{res} data points, and $r_i(x) = y_i - \varphi(t_i; x), \quad i = 1, \ldots, n_{\text{res}}$ is the i^{th} residual, equal to the difference between the observed and predicted values of the independent variable at time t_i, denoted y_i and $\varphi(t_i; x)$ respectively.

The loss function χ has desirable properties such as being bounded from below, and increasing with $|r_i(x)|$. Summing over all data points then, the objective function will be small when the model fits the whole dataset well, which is what we want.

There are plenty of choices for function χ, and one important consideration is robustness. A robust loss function is one which doesn't get thrown off easily by outliers in the data.

NAG's Generalized Nonlinear Data Fitting solver (`e04gn` in the NAG Library) makes it easy to choose among several loss functions, including l_1-norm, l_2-norm, and $\arctan$. We used this solver with these options to demonstrate the results produced by loss functions of varying robustness.

Single-Outlier Example

To investigate the robustness aspect, we'll start with a toy dataset of 21 points generated from $\sin(t)$ with an outlier at $t = 1.5$, which is generated by $5\sin(t)$.

We will fit it with a model,

$$\varphi(t; x) = x_1 \sin(x_2 t)$$

using first the l_2 loss function, then the l_1 loss function. The starting point for each solve is $x = (2.1, 1.4)$.

l_2-Norm Loss Function

The l_2-norm is one of the most common loss functions. This forms the problem as a least squares regression,

$$\underset{x \in \mathbb{R}^2}{\text{minimize}}\ f(x) = \sum_{i=1}^{21} r_i(x)^2.$$

l_1-Norm Loss Function

Using l_1-norm loss gives us the problem,

$$\underset{x \in \mathbb{R}^2}{\text{minimize}}\ f(x) = \sum_{i=1}^{21} |r_i(x)|,$$

which is more robust against outliers. This means if some large portion of the data is well-fitted by some solution x^*, there is likely to be a local minimum very close to x^* which is relatively undisturbed by the remaining data that is outlying to the solution x^*.

Comparison

The results in Figure 1 show the model fitted with the l_1 loss function was clearly better than the model fitted with the l_2 loss function where outliers contribute heavily to the objective function and search direction.

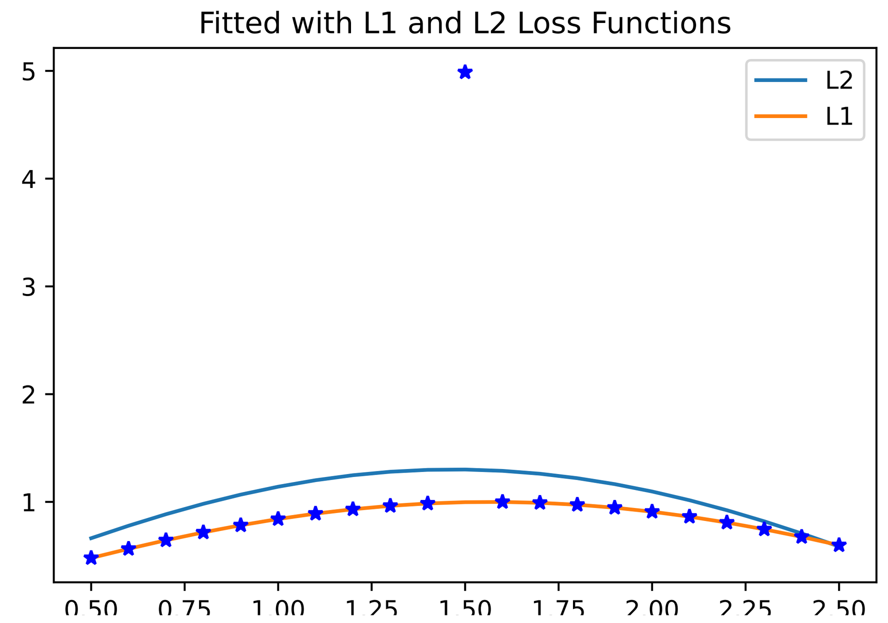


Figure 1

The Trade-Off of a Loss Function

There is a danger in choosing a very robust loss function. During an iterative optimization process, a loss function which is robust against outliers will usually prefer the data which is close to the current model. This means that if the algorithm finds local minima of the objective function, the search can fall into a local minimum when the model fits some subset of the data very well but fits the majority of the data very badly.

To illustrate this, we will fit the same model to a new dataset generated by $5\sin(t)$, with 3 data points on each end generated by $\sin(t)$, using l_1, l_2, and $\arctan$ loss functions (Figure 2).

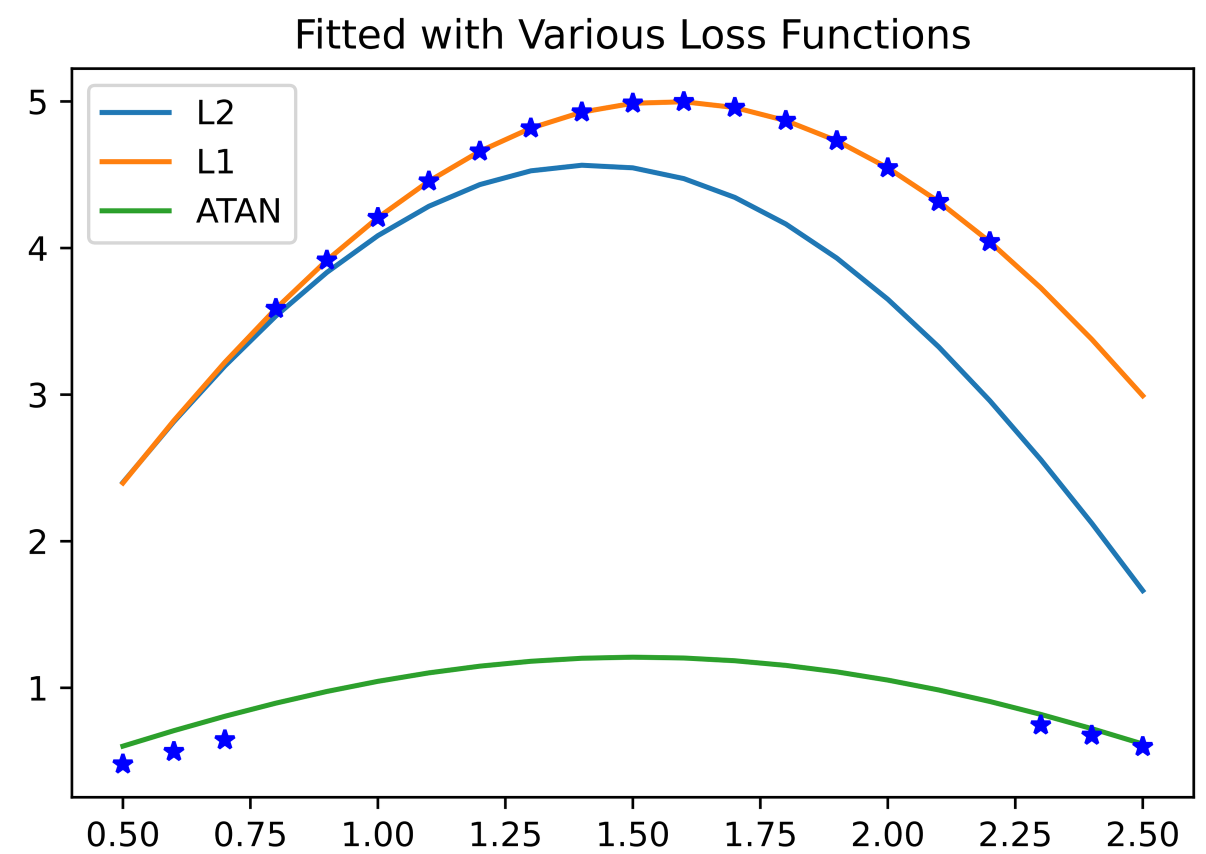
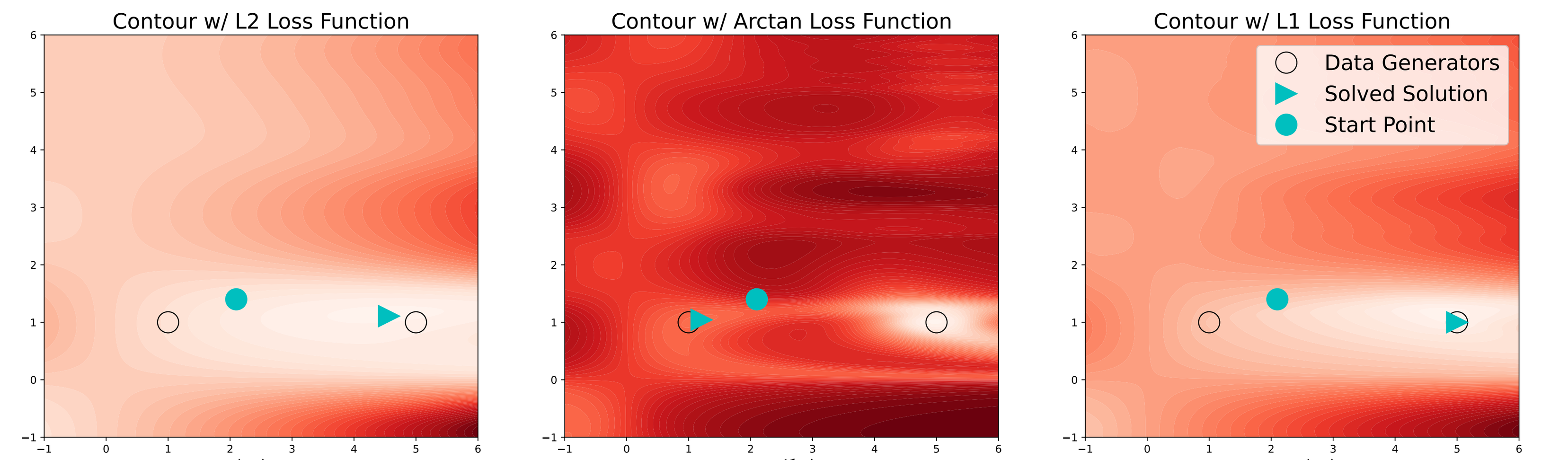


Figure 2

Contour Plots

In the contour plots below, the black circles represent the parameters used to generate the data, the cyan circle represents the starting point for the solver, and the cyan wedges represent the optimized solution found by the solver.



(a) (b) (c)

With the l_2-norm in (a), the outliers generated by $\sin(t)$ have pulled the optimal solution away from $x = (5, 1)$. The contour plot for l_2-norm loss indicates that we don't have to worry too much about what starting point to use, since there are no local minima in the region displayed, other than global best solution.

The behaviour of the solver is quite different when using an extremely robust loss function like $\arctan$ loss, which looks like

$$\underset{x \in \mathbb{R}^2}{\text{minimize}}\ f(x) = \sum_{i=1}^{21} \arctan(r_i(x)^2)$$

There are eight local minima in the contour plot for $\arctan$ (b), with seven of them being substantially worse solutions than the global minimum, and it is one of these we've converged to. In this case, the initial estimate led to a model that only fit a small portion of the data very well.

In the l_1-norm contour plot (c), there are still a few local minima that do not correspond to the optimal solution, but the starting point of $x = (2.1, 1.4)$ still converges to the global minimum, which lies at $x = (5, 1)$, meaning the part of the dataset generated from $\sin(t)$ is effectively being ignored. From the plots of the loss functions, we can see that l_1-norm loss is more robust than l_2-norm loss but less so than $\arctan$ loss.

Conclusion

Your choice of loss function can affect your model's sensitivity to outliers, populate the search space with more local minima, and/or cause more sensitivity to the starting point. Using the `e04gn` solver, it is easy to try different loss functions while setting up your data fitting problem, ensuring you spend your time solving for the *right* optimal solution.

Find more examples *with* source code at:

github.com/numericalalgorithmsgroup