Prediction Intervals and Confidence Regions for Symbolic Regression Models based on Likelihood Profiles

F. O. de Franca* and G. Kronberger†
*Center for Mathematics, Computation and Cognition (CMCC), Heuristics, Analysis and Learning Laboratory (HAL),
Federal University of ABC, Santo Andre, Brazil (e-mail: folivetti@ufabc.edu.br)
†Josef Ressel Center for Symbolic Regression,
Heuristic and Evolutionary Algorithms Laboratory (HEAL),
University of Applied Sciences Upper Austria,
Softwarepark 11, 4232 Hagenberg, Austria

***Abstract*—Uncertainty quantification (UQ) in regression models helps with the decision-making process by enabling the user to understand the reliability and limitations of a model. This is well established for linear models and has been extended to nonlinear models using linear approximations. However, such approximations can be inaccurate depending on the sample size and the nonlinearity of the likelihood function. A more accurate alternative is the likelihood profile method, which takes into consideration the nonlinearity of the model and correlated parameters. While this method is well known in nonlinear regression literature, it remains underutilized in symbolic regression (SR) for quantifying the uncertainties of selected models. An additional challenge is the calculation of prediction intervals (PI) using likelihood profiles, which requires algebraic manipulation of the symbolic expression, which is often done manually in practice. In this work, we explain three different methods to calculate the profile likelihood of symbolic expressions and, additionally, we propose an algorithm for the computation of prediction intervals.**

***Index Terms*—Nonlinear Regression, Symbolic Regression, Uncertainty Quantification, Prediction Interval, Confidence Interval**

I. Introduction

Whenever we build a regression model from data, we are subject to different sources of uncertainties such as measurement noise, choice of explanatory variables, or sampling bias. The inferences made by regression models are subject to these uncertainties, having a quantification of such uncertainties can help with a decision process. Imagine a regression model that makes inference about the precipitation at a certain region for a given period of time. Knowing how much it will rain can help warning people living areas endangered by landslides. Let us say that the decision threshold to move people to shelters is a value of 100mm of rain or more. If the model predicts 90mm of rain, the decision would be to keep everyone at their homes. But this number accounts only for the expected values ignoring all the uncertainties involved. By taking the uncertainties into account, we could have reported the same expected value of 90mm but with a confidence interval (CI) $[75, 105]$, for example. With this information we can make a more informed decision so that, if we decide not to move them out of their homes, we can at least keep personnel on alert.

The same can be said about the parameter estimates; knowing the involved uncertainties can help to interpret the model. Besides inference, another important task of regression analysis is association. In this case, we are interested in understanding the association of a given variable with the output of the system. Suppose we create a linear regression model that investigates the association of education level and height with salary range. If the associated parameters are 1.3 and 0.1 for the respective variables, we can say that increasing education level by 1 will increase the salary range by 1.3 units, while increasing your height by one unit, increases the salary range by 0.1 units. If we accept these values as it is, we would argue that both variables have a positive effect on salary. But, let us say the confidence interval for these coefficients are $[1.0, 1.6]$ and $[-0.1, 0.2]$. In this case, we can be confident that the education level does have a positive effect on salary, but the effect of height may have been simply a random artifact or sample bias.

As can be seen from these examples, having a confidence interval for the predictions and the numerical coefficients can help to understand models and make better decisions. The calculation of confidence intervals is well-established for linear regression with Gaussian likelihood and readily available in most software packages. For nonlinear models, given we know the regression function and its partial derivatives, we can also calculate a linear approximation of the confidence interval [1]. This is frequently called the "delta method". Bates and Watts described an even better approach which use the profile t function [1], also called likelihood profile, to calculate more accurate confidence intervals, with the cost of having to apply a nonlinear solver multiple times. While not as widespread as in linear regression toolboxes, there are implementations of both the linear approximation and likelihood profiles for

generalized linear or nonlinear regression models[1].

In traditional nonlinear regression, first the user proposes the regression function and then the coefficients are fitted to the data according to the likelihood function. Symbolic regression (SR) [2, 3], on the other hand, searches for the optimal functional structure, thus not being limited to a predetermined model. If the returned model is differentiable, SR models can be treated as any nonlinear model and, therefore, it is possible to quantify the uncertainties using the same methods as in the regression analysis literature.

The concept of likelihood profiles for nonlinear regression is well-known in the statistics community but the algorithms published in [1]. Additionally, the algorithm for profile-based prediction intervals is not described at all in [1]. SR literature is mostly focused only on pointwise predictions, with only a few papers acknowledging the need for uncertainty quantification, and even less using likelihood profiles [3]. As a consequence SR implementations usually do not provide confidence or prediction intervals at all or use various heuristics such as lower-upper bound estimation (LUBE) [4].

This paper contains two contributions: first, we give clear pseudo-codes for the calculation of likelihood profiles and the approximate contours for pairwise parameter confidence regions. Second, we provide an algorithm for algebraic manipulation that enables the use of profile likelihood to calculate the prediction intervals of symbolic expressions.

The paper is organized as follows: we first introduce the basic definitions of the calculation of Sec. II details the calculation of CI in linear regression, the linear approximation for nonlinear models and the use of the likelihood profile for nonlinear intervals. In Sec. III we provide the requirements and algorithmic details of the profile likelihood calculation for symbolic regression models. In Sec. IV we provide an illustrative example of the method followed by a discussion in Sec. V. Finally, in Sec. VI we discuss limitations of the approach and some additional insights.

II. Confidence Intervals for Nonlinear Models

The following is mainly based upon [1], we reproduce this for completeness. Given a dataset formed by n samples of p independent variables $\mathbf{X} \in \mathbb{R}^{n\times p}$ and corresponding dependent (also called target) measurements $\mathbf{y} \in \mathbb{R}^n$, a linear regression model parameterized by numerical coefficients $\beta \in \mathbb{R}^p$ is given by:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \tag{1}$$

where ϵ is called residual and corresponds to the error term associated with the uncertainty of the collected data. The linear model assumes that the residuals are i.i.d. normally distributed with zero mean and that the relationship between the independent and dependent variables is linear.

The $X\beta$ part of the model is called the expectation function and it is related to the inference of the expected value of the target variable ($\mathbf{E}[\mathbf{y}] = \mathbf{X}\beta$). Specifically for the linear model, $\mathbf{X}$ is also called the derivative matrix, as the partial derivative of $\mathbf{y}$ w.r.t. β is $\mathbf{X}$. To determine the values of β we can solve the least square problem by minimizing the residual sum of the squares (SSR) to obtain the optimal $\hat{\beta}$:

$$SSR(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2, \tag{2}$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3}$$

The residual mean square (s^2) or variance estimate is calculated based on $n-p$ degrees of freedom:

$$s^2 = \frac{\mathrm{SSR}(\hat{\beta})}{n-p}. \tag{4}$$

The $1-\alpha$ confidence interval of a coefficient β_i can be calculated using the upper $\alpha/2$ percentile for a t-distribution with $n-p$ degrees of freedom:

$$\hat{\beta}_i \pm \mathrm{se}(\hat{\beta}_i)\, t(n-p, \alpha/2), \tag{5}$$

$$\mathrm{se}_i = s\sqrt{((\mathbf{X}^T\mathbf{X})^{-1})_{ii}}, \tag{6}$$

where A_{ii} represents the i-th diagonal element of a matrix A. Similarly we can calculate the $1-\alpha$ confidence interval of the prediction at a new point $\mathbf{x}$ as:

$$\mathbf{x}^T\hat{\beta} \pm s\sqrt{\mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}}\, t(n-p, \alpha/2). \tag{7}$$

When working with nonlinear models, we are interested in models:

$$\mathbf{y} = f(\mathbf{X}, \theta) + \epsilon, \tag{8}$$

where θ corresponds to the numerical coefficients of the model and we assume a probability distribution for ϵ (often a Gaussian distribution). This allows us to estimate the optimal parameters $\hat{\theta}$ by optimizing the maximum likelihood with a nonlinear optimization method. These algorithms use the gradient information provided by the Jacobian J calculated for the residual ϵ_k w.r.t. a certain θ

$$J_{kj} = \frac{\partial \epsilon_k}{\partial \theta_j},\ k = 1 \ldots n. \tag{9}$$

A. Linear approximation

Now, given the QR decomposition of J evaluated at $\hat{\theta}$, we can calculate the linear approximation for the standard error se_i of the i-th parameter as the length of the i-th row of R^{-1}, and the residual error rse for new data points

$$\mathrm{se}_i = s\sqrt{\sum_j (R^{-1})_{ij}^2}$$

$$\mathrm{rse}_k = s\sqrt{\sum_j (JR^{-1})_{kj}^2},$$

where J is the Jacobian for these data points when using $\hat{\theta}$ and R^{-1} estimated from the training set.

The confidence intervals of $\hat{\theta}$ are then calculated as $\theta_i \pm \mathrm{se}_i\, t(n-p, \alpha/2)$. Likewise, the confidence interval of the

[1]The R nls() function is an important example.

predictions for any point $\mathbf{x}$ and the prediction interval of any data point can be calculated as:

$$f(\mathbf{x}, \hat{\theta}) \pm \text{rse}\ t(n-p, \alpha/2)$$
$$f(\mathbf{x}, \hat{\theta}) \pm (\text{rse} + s)\, t(n-p, \alpha/2),$$

since these are linear approximations of the true intervals, they can be "extremely misleading", as noted in [1].

B. Likelihood profile

The likelihood profile takes the correlation between parameters in consideration to build a profile of each parameter. The main idea is that, assuming θ_i as the parameter of interest and $\theta_{j \neq i}$ the nuisance parameters, we fix θ_i at different values and for each value we optimize only the nuisance parameters with the likelihood function, storing the new optimal value $\tilde{\theta}$ and the profile corresponding value of the profiling function $\tau(\theta_i)$

$$\tau(\theta_i) = \text{sign}(\theta_i - \hat{\theta}_i)\sqrt{2L(\tilde{\theta}) - 2L(\hat{\theta})}, \tag{10}$$

where L is the log-likelihood. In contrast to Eq. 6, the CI of θ_i using the profile is the supremum and infimum of the set

$$\{\theta_i \mid -t(n-p, \alpha/2) \leq \tau(\theta_i) \leq t(n-p, \alpha/2)\}. \tag{11}$$

If the closed-form of $\tau(\theta_i)$ is known, the CIs obtained using likelihood profile would be exact [1]. However, because the closed-form expression for $\tau(\theta_i)$ is intractable, this set must be estimated using an approximation algorithm, which does not hold this guarantee of being exact [5].

A *naive* approximation method is described in [1], and consists of evaluating $\tau(\theta_i)$ for different $\hat{\theta}_i + \delta k$, where δ is the direction of the step and k is the step size. The step size is updated at every iteration with an attempt to be closer to the boundary $t(n-p, \alpha/2)$, thus minimizing the number of samples needed to find the boundaries. The direction is set as $\pm \text{se}/8$, depending whether the profile is moving towards the infimum or supremum, and the initial step size is set to $t = 1$. At every iteration j, the tuple $(\tilde{\theta}^{(j)}, \tau(\theta_i)^{(j)})$ is stored, where $\tilde{\theta}$ is estimated by optimizing the nuisance variables while fixing the value of θ_i. Finally, the step size k is updated according to

$$t \leftarrow t + \min(4, \max(\left|\frac{\tau(\theta_i)s^2}{\text{se}_i\, \epsilon^T \tilde{J}_{_i}}\right|, 1/16)) \tag{12}$$

where $\tilde{J}$ is the Jacobian at $\tilde{\theta}$. These steps are iterated until a maximum of 30 iterations or $|\tau(\theta_i)| > \sqrt{F(1-0.01, p, n-p)}$ (i.e., the limit of τ for a 99% confidence interval), where F is the F-distribution. This procedure is described in details in Alg. 1 for a certain parameter i and direction δ. One caveat with this procedure is that if the optimization procedure finds a higher likelihood value than the value for $\hat{\theta}$, the procedure must be restarted with the new obtained values. This can happen if the optimization method was interrupted before convergence (i.e., reached maximum iteration) or if the disturbance caused by δ leads to a new basin of attraction with a better optimum.

The main disadvantage of this method is the need to repeat the optimization process for every sampled θ_i, however,

Algorithm 1 Likelihood profile algorithm as described in [1], $\mathbb{1}_i$ is a vector with 1 at position i and 0 otherwise.

```
function PROFILE(i, θ̂, f, J, δ)
    P ← {}, t ← 1, it ← 1, θ̃ ← θ̂
    while it ≤ 30 and |τ(θ̃)| ≤ τ_max do
        θ̃ ← θ̂ + 1_i δt
        θ̃ ← optimize(θ̃, f, J̃)
        t ← update(t, τ(θ̃))                    ▷ Eq. 12
        P ← P ∪ {(θ̃, τ(θ̃))}
        it ← it + 1
    end while
    return P
end function
```

since it returns a set of sampled values, it is possible to interpolate τ using any nonlinear interpolation method. This interpolation can be used to create a profile plot depicting the nonlinearity behavior of the intervals. Bates and Watts also sketch an algorithm for the approximation of pairwise parameter confidence regions based on the likelihood profiles [1], described in Alg. 2. The contours produced by this algorithm are good approximations for the true confidence regions when the parameter estimates are well-determined and close to ellipsoid as shown in the example plots (Fig. 2).

A faster approximation of the profile trace is given in [6] where the problem is formulated as an ordinary differential equation (ODE) of the form

$$\begin{pmatrix} -W(\theta_c) & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\theta}_c \\ \mu_c \end{pmatrix} = \begin{pmatrix} -\gamma \dot{l}(\theta_c) \\ 1 \end{pmatrix}, \tag{13}$$

where θ_c is the set of values approximating the true profile trace of θ, W is the approximation of the Hessian matrix at that value, $\dot{l}$ is the gradient of the likelihood function, and μ is the Lagrange multiplier of the problem formulation acting as a self-regularization of the trajectory. By using an iterative ODE solver (e.g., Runge–Kutta), the bottom part of the first matrix will ensure a unitary step for θ while adjusting the direction and step size with the Hessian and gradient information. Using this method it becomes unnecessary to reoptimize the likelihood function for every proposed step, being much faster than the traditional approach.

If only the endpoints of the profile trace are needed, it is possible to make the calculation even more computationally efficient using the following constrained optimization (for the upper bound):

$$\max \quad \theta_i \tag{14}$$
$$\text{s.t.} \quad l(\theta) - l^*_\alpha \leq 0, \tag{15}$$

where $l(\theta)$ is the log-likelihood value at that point, $l^*_\alpha = l(\hat{\theta}) + \tau_{max}$. This can be solved with the augmented Lagrangian [7] transforming the constrained optimization function into an unconstrained optimization problem with penalization.To find the lower bound, the objective-function is changed to a minimization problem.

Algorithm 2 Algorithm for the calculation of approximated contour plots for the pairwise parameter confidence regions. This algorithm assumes the existence of a function called *Spline* that returns a cubic spline function based on the input points. *PeriodicCubicSpline* returns the spline with period 2π. $F(1-\alpha, p, n-p)$ is the critical value for the F-distribution with p and $n-p$ degrees of freedom, λ_i is the spline interpolating the function $\tau(\theta_i)$, g_i^j is the spline interpolation mapping $\tau(\theta_j)$ to the inverse cosine of $\lambda_i(\Theta_j)$.

function PROFILECONTOUR($i, j, \Theta, T, \alpha, steps$)
  $\tau_{scale} \leftarrow \sqrt{p\,F(1-\alpha, p, n-p)}$
  $g_i^j \leftarrow$ SPLINE($\tau_j, \arccos(\lambda_i(\Theta_j)/\tau_{scale})$)
  $g_j^i \leftarrow$ SPLINE($\tau_i, \arccos(\lambda_j(\Theta_i)/\tau_{scale})$)
  $\text{ang} \leftarrow [(0, g_i^j(1)), (g_j^i(1), 0), (\pi, g_i^j(-1)), (g_j^i(-1), \pi)]$
  for $k = 0 \ldots 3$ **do**
    $d_k \leftarrow \text{ang}_{k0} - \text{ang}_{k1}$
    $a_k \leftarrow \text{sign}(d_k)(\text{ang}_{k0} + \text{ang}_{k1})/2$
    $d_k \leftarrow \text{abs}(d_k)$
  end for
  $a_4 \leftarrow a_0 + 2\pi, d_4 \leftarrow d_0$
  $p \leftarrow$ PERIODICCUBICSPLINE(a, d)
  for $k = 1 \ldots \text{steps}$ **do**
    $x \leftarrow 2k\pi/(\text{steps} - 1) - \pi$
    $\tau_{ik} \leftarrow \cos(x + p(x)/2)\tau_{scale}$
    $\tau_{jk} \leftarrow \cos(x - p(x)/2)\tau_{scale}$
    $\theta_{ik} \leftarrow \lambda_i(\tau_{ik}), \theta_{jk} \leftarrow \lambda_j(\tau_{jk})$
  end for
  return θ_i, θ_j
end function

C. Profile-based prediction intervals

Likelihood profiles can also be used to calculate continuous asymmetric prediction intervals of a new data point x^* exploiting the same algorithm described in Alg. 1. The idea is to rewrite the symbolic expression such that one of the parameters is replaced by the output of the model at the evaluation point x^*. As a simple example, suppose the nonlinear model is described as:

$$\hat{y} = f(x) = \hat{\theta}_0 e^{\hat{\theta}_1 x}. \tag{16}$$

To calculate the prediction interval for $\hat{y}$ at point x^*, we first rearrange the equation so that one of the coefficients is the output. Let us pick the first coefficient, then we will have:

$$\hat{\theta}_0 = \hat{y} e^{-\hat{\theta}_1 x}. \tag{17}$$

We then rename $\hat{y}$ in the above expression as our new parameter θ_0' which represents the output of the model in point x^* and replace in the expectation function:

$$f(x)' = \theta_0' e^{-\hat{\theta}_1 x^*} e^{\hat{\theta}_1 x}. \tag{18}$$

The value for the new parameter θ_0' is calculated via Eq. 16 for the point x.

Now, we apply Alg. 1 with the re-parameterized model $f(x)'$ for the new parameter θ_0' after calculating the standard errors for the re-parameterized model whereby we use the training set X for the optimization.

These steps ensure that one of the model parameters equals the prediction $\hat{y}_i$ at the point x^* which can be easily verified in Eq. 18. The likelihood profile for this parameter provides the nonlinear prediction interval in this point. The re-parameterization and profile calculation has to be repeated for each point for which we evaluate the prediction interval.

III. Likelihood Profiles for Symbolic Regression

The algorithms described in the previous section apply to any regression model whose parameters are estimated using likelihood maximization. Consequently, this approach extends to models produced by SR, provided they apply such optimization procedure.

Another requirement for this methodology is parameter identifiability: the model must not contain linearly dependent parameters. Since models generated by GP algorithms often do not fulfill this requirement [8], algebraic simplification is often necessary before computing the likelihood profile.

Alg. 3 details the steps to generate the inverted symbolic model. This algorithm starts by building a new symbolic tree where the parameter θ_0 is isolated becoming the output of the function. It achieves this by traversing the original tree in pre-order and applying algebraic inversions to the branches containing θ_0:

- If the node is θ_0, it applies any inverse link function required by the likelihood (e.g., $\log$ for Poisson).
- If the node is a non-terminal, it applies the corresponding inverse operator (e.g., $\exp$ for $\log$) or the left or right inversion for binary operators, depending on the position of θ_0.
- If it is an input variable, it is evaluated at x^*, generating a constant node.

Finally, the generated inverse tree replaces the θ_0 node in the original structure, creating the exact mathematical expression required for the profile likelihood algorithm. Notice that every non-terminal node that is an ancestor of θ_0 should allow the inverse operation.

IV. Illustrative Example

To demonstrate the profile likelihood, we will resort to the same example as described in [1] for the BOD dataset, a simple dataset with only 6 samples and a single feature: $x = [1, 2, 3, 4, 5, 7], y = [8.3, 10.3, 19.0, 16.0, 15.6, 19.8]$, and the following regression model:

$$19.143(1 - e^{-0.5311x}) \tag{19}$$

where $\theta = [19.143, -0.5311]$ are the adjustable parameters.

We used the implementation of the delta method and profile likelihood available in *srtools* [2] setting the number of iterations for the optimization method to 100 and using the mean squared

[2] https://github.com/folivetti/srtree/tree/main/apps/srtools

Algorithm 3 Algorithm for symbolic rewriting of expressions, n^θ indicates a branch that has θ_0.

```
function REWRITE(t)
    t' ← INVERT(t)
    return REPLACE(t, t')
end function

function INVERT(t)
    switch t
        case x_i → EVAL(x_i)
        case θ_0 → l^{-1}(θ_0)
        case θ_i → θ_i
        case g(n) →
        switch n
            case n^θ → inverseOf(g, INVERT(n))
            case n → g(n)
        end switch
        case op(l, r) →
        switch (l, r)
            case (l^θ, r) → inverseLeft(op, INVERT(l), r)
            case (l, r^θ) → inverseRight(op, l, INVERT(r))
            case _ → op(l, r)
        end switch
    end switch
end function

function REPLACE(t, t')
    if t' = θ_0 then
        return t
    else
        t'(REPLACE(c) for c ∈ children-of(t'))
    end if
end function
```

error as the loss function. All the intervals are calculated with a significance level $\alpha = 0.05$. For comparison we implemented the model in Turing.jl setting the prior parameters to the maximum likelihood estimate to allow MCMC sampling of the parameter distribution and the predictive distribution. We collected 10000 samples and used 5000 for burn in. For the Bayesian results we report the interval from the 2.5% to 97.5% percentiles.

Table I compares the CIs of the parameters obtained by the different methods. We can see from this table that the delta method fails to capture the asymmetry of the intervals as in the profile likelihood. While the profile likelihood interval for θ_0 is wider due to this asymmetry, the interval for θ_1 is much narrower. Comparing the different methods, only the constrained optimization approach delivers a slightly different interval, though still closer than the delta method.

Fig. 1 shows the likelihood profiles for each parameter. In this plot the x-axis is the parameter value and the y-axis is the corresponding value of τ, the value of τ is related to the degree-of-confidence so, in this particular plot, the values of θ that corresponds to a $-t(n-p, 0.95/2) < \tau < t(n-p, 0.95/2)$ are the values for a 95% confidence interval.

TABLE I
CIs OF THE PARAMETERS AS CALCULATED BY THE DIFFERENT METHODS.

	θ_0	θ_1
delta	$[13.53, 24.75]$	$[-0.99, -0.07]$
Bates	$[15.58, 26.44]$	$[-1.08, -0.24]$
ODE	$[15.58, 26.44]$	$[-1.08, -0.24]$
Constr.	$[15.97, 24.97]$	$[-0.97, -0.26]$
Bayesian	$[15.37, 22.33]$	$[-1.81, -0.33]$

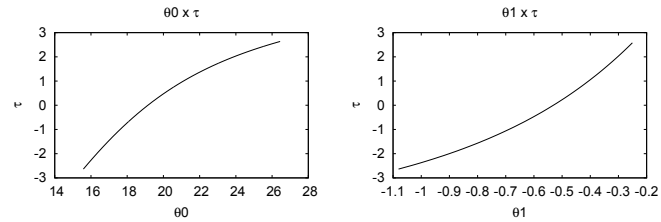

Fig. 1. Likelihood profiles for Eq. 19. We can see from this plot that, for this model, the parameters strongly deviate from linearity.

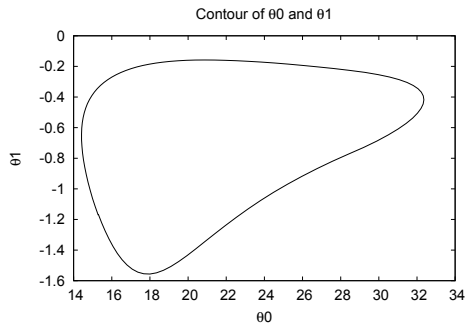

Fig. 2. Pairwise approximate parameter confidence regions for Eq. 19 and $\alpha = 0.05$.

TABLE II
PREDICTION INTERVALS AND THE WIDTH (Δ) OF EACH SIDE OF THE PREDICTION. IN THIS TABLE y IS THE POINTWISE MEAN PREDICTION, y^l IS THE LOWER BOUND AND y^u IS THE UPPER BOUND FOR $\alpha = 0.95$.

y^l	y	y^u	Δ_l	Δ_u	y^l_{Bay}	y_{Bay}	y^h_{Bay}
4.64	7.88	13.08	3.24	8.43	6.04	8.63	12.51
8.55	12.52	16.50	3.96	7.94	10.29	13.03	15.96
11.74	15.25	18.38	3.51	6.64	13.14	15.43	17.63
13.80	16.85	19.89	3.04	6.09	14.68	16.79	18.75
14.13	17.79	21.34	3.65	7.20	15.33	17.58	19.64
14.22	18.67	24.29	4.45	10.07	15.57	18.34	20.86

Fig. 2 shows the pairwise likelihood of each pair of parameters. In this plot we can see how the parameters interact. For example, given the nonlinearity of the parameters, for a given value of θ_0, the interval of θ_1 is narrower than the ones given by the independent analysis (as in Figure 1). When the parameters are close to linear, the pairwise plots will be ellipsoid, larger correlations leading to elongated ellipses. Strong nonlinearity leads to "banana-shaped" contours.

In Fig. 3 we can see that both types of prediction intervals for Eq. 19 contain all of the data points. As it is the case for

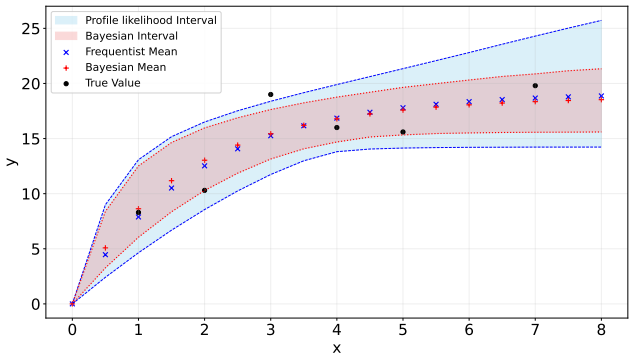

Fig. 3. Prediction intervals for the BOD dataset using profile likelihood. The asymmetry is evidenced in Table II.

the CIs, the profile PIs are not necessarily symmetrical, as can be observed in Table II.

V. Discussion

From the above example, we can see how easily CIs and PIs can be calculated for SR models. The main requirement is that the SR model is algebraically simplified and redundant parameters are removed. The linear approximation is easy to calculate and very accurate when model parameters are well-estimated, i.e. for datasets with low noise or the parameter effects are close to linear. For models with strong nonlinearity the likelihood profile can give much better CIs and PIs when compared to linear approximations.

The likelihood profile can also reveal model issues in particular unbounded intervals for parameter estimates. The pairwise plots can visually aid the practitioner to determine the relationship between two parameters.

Whenever these plots reveal a problem with the model, we can try to fix it finding a new model, rewriting the expression or refitting the parameters. This can be done manually or computer-aided with computer algebra systems because SR uses a symbolic model representation.

VI. Conclusion

In this paper we have proposed the use of likelihood profiles for estimating the uncertainties of the parameters and predictions of nonlinear symbolic regression models. Unlike the commonly used linear approximation, the likelihood profile returns more accurate intervals especially for models with strong nonlinearity.

We gave clear pseudo-code for the algorithms originally described by Bates and Watts [1], and additionally describe how the likelihood profiles can be used for prediction intervals, which is only hinted at briefly in the original source and not explored in the literature.

While the original implementation requires multiple re-optimization of the parameters, leading to a high computational cost, there are faster alternatives that alleviate this cost when only the endpoints of the intervals are required. When attained to the original method, the profiles can be used to generate supporting plots that can identify issues with the model or bring a better understanding of the overall behavior of the studied system.

In conclusion, the likelihood profile method is a valuable statistical technique for analysis and validation of SR models that can help the practitioners to extract additional information about the model, inspect the validity, fix ill-conditioning, and understand the limitations of the model.

Acknowledgment

F.O.F. is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) grant 301596/2022-0. G.K. acknowledges support by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Federal Ministry for Labour and Economy, and the regional government of Upper Austria within the COMET project ProMetHeus (904919) supported by the Austrian Research Promotion Agency (FFG).

References

[1] D. Bates and D. Watts, *Nonlinear regression analysis and its applications*, ser. Wiley series in probability and mathematical statistics. Wiley, 1988.

[2] W. La Cava, P. Orzechowski, B. Burlacu, F. O. de Franca, M. Virgolin, Y. Jin, M. Kommenda, and J. H. Moore, "Contemporary symbolic regression methods and their relative performance," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[3] G. Kronberger, B. Burlacu, M. Kommenda, S. M. Winkler, and M. Affenzeller, *Symbolic Regression*. Chapman & Hall / CRC Press, 2024.

[4] K. B. Rebuli, M. Giacobini, N. Tallone, and L. Vanneschi, "A preliminary study of prediction interval methods with genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 530–533. [Online]. Available: https://doi.org/10.1145/3520304.3528806

[5] S. L. Quinn, T. J. Harris, and D. W. Bacon, "Notes on likelihood intervals and profiling," *Communications in Statistics-Theory and Methods*, vol. 29, no. 1, pp. 109–129, 2000.

[6] J.-S. Chen and R. I. Jennrich, "Simple accurate approximation of likelihood profiles," *Journal of Computational and Graphical Statistics*, vol. 11, no. 3, pp. 714–732, 2002. [Online]. Available: http://www.jstor.org/stable/1391121

[7] A. R. Conn, N. I. Gould, and P. Toint, "A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28, no. 2, pp. 545–572, 1991.

[8] F. O. de Franca and G. Kronberger, "Reducing overparameterization of symbolic regression models with equality

saturation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2023, pp. 1064–1072.