Bayesian Model Selection for Reducing Bloat and Overfitting in Genetic Programming for Symbolic Regression

Geoffrey F. Bomarito [1] Patrick E. Leser [1] Nolan C.M. Strauss [2] Karl M. Garbrecht [2] Jacob D. Hochhalter [2]

[1]NASA Langley Research Center, [2]University of Utah

GECCO 22, Boston, MA, July 9-13, 2022

Abstract

When performing symbolic regression using genetic programming, overfitting and bloat can negatively impact generalizability and interpretability of the resulting equations as well as increase computation times. A Bayesian fitness metric is introduced and its impact on bloat and overfitting during population evolution is studied and compared to common alternatives in the literature. The proposed approach was found to be more robust to noise and data sparsity in numerical experiments, guiding evolution to a level of complexity appropriate to the dataset. Further evolution of the population resulted not in overfitting or bloat, but rather in slight simplifications in model form. The ability to identify an equation of complexity appropriate to the scale of noise in the training data was also demonstrated. In general, the Bayesian model selection algorithm was shown to be an effective means of regularization which resulted in less bloat and overfitting when any amount of noise was present in the training data.

Method

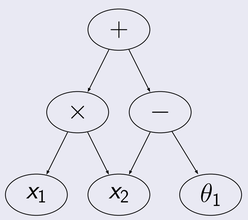

Figure 1: Acyclic graph representation of the equation $f(\mathbf{x}, \boldsymbol{\theta}) = x_1 x_2 + x_2 - \theta_1$ with complexity of 6.

Genetic Programming for Symbolic Regression (GPSR) with the Python package `Bingo`[a] involves learning an acyclic graph (Fig. 1) that minimizes a fitness metric such as root mean squared error (RMSE) over a set of training data. In standard form, `Bingo` assumes noise-free data though this is rarely the case in reality.

[a]github.com/nasa/bingo

In this work, a Bayesian fitness metric is developed for `Bingo` that reduces bloat through natural penalization of complexity and reduces overfitting by modeling noise explicitly rather than trying to fit a function to the data deterministically.

- Uncertainty in model parameters, θ, is quantified using Bayesian inference for each candidate equation.
- With no knowledge about the support and relative probability of θ, the use of improper uniform priors, $U(-\infty, \infty)$, is necessary.
- These priors invalidate use of Bayes' factor for model selection.
- Instead, **Fractional Bayes' Factor (FBF)** [1] is utilized ($B_F = q_1/q_2$, where q_i is the **normalized marginal likelihood (NML)** for model i).
- Sequential Monte Carlo (SMC) as implemented in the Python package `SMCPy`[b] is uniquely suited for efficient computation of FBF.

Two popular selection algorithms are investigated with the new metric: deterministic crowding (DC) [2] and probabilistic crowding (PC) [3].

[b]github.com/nasa/smcpy

Numerical Experiment

In this experiment, datasets are generated using:

$$y_i = 2\sin(x_{0,i}) + 3 + \epsilon_i \tag{1}$$

where $x_{0,i}$ is independently sampled from a uniform distribution, $x_{0,i} \overset{iid}{\sim} U(0, \frac{3\pi}{2})$, and ϵ_i represents additive noise that is Gaussian with standard deviation σ. For each $\sigma \in \{0, 0.05, 0.1, 0.3, 0.5, 0.7, 1.0\}$ 50 training datasets of 20 points are generated (each with independent samples of x_0 and ϵ).

Four GPSR algorithms are tested:

1. A base GPSR algorithm: DC and RMSE.
2. A GPSR algorithm using DC and NML.
3. A GPSR algorithm using PC and RMSE.
4. The FBF-based GPSR algorithm: PC and NML.

Results

In all figures, lines and shaded areas represent mean values and 95% confidence intervals from 50 repetitions, respectively.

- All algorithms improve fitness with evolution, though the steady decline for RMSE is a sign of overfitting (Fig. 2).
- Bloat is reduced by using NML rather than RMSE (Fig. 3).
  - The number of parameters and complexity continuously increase for RMSE algorithms (bloat).
  - NML saturates at some number of parameters and complexity, and complexity even tends to decrease with further evolution.

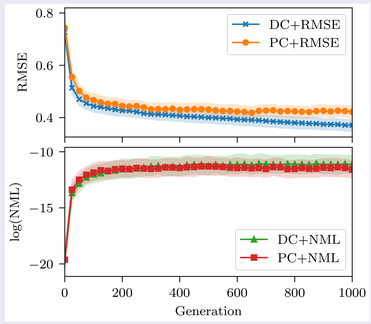

Figure 2: Fitness during evolution, $\sigma = 0.5$.

- Adaptation to σ is an attractive property of PC and NML (Fig. 4).

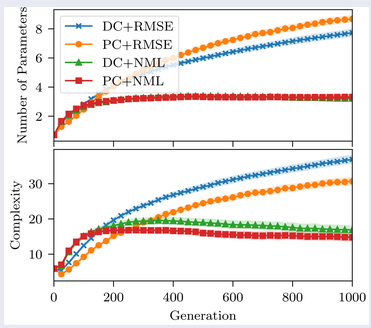

Figure 3: Complexity and parametric dimension during evolution, $\sigma = 0.5$.

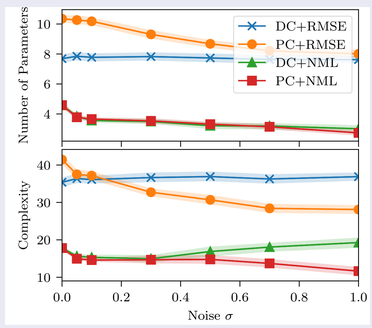

Figure 4: Complexity and parametric dimension for varying σ (1000 generations).

Results (Continued)

To assess generalizability, 1000 test data points were generated using Equation (1) for each σ. Performance was quantified by RMSE, meaning σ serves as a *lower bound* [c] for comparison.

[c]Available GPSR operators were not sufficient to learn the data-generating equation precisely.

- Over generations (Fig. 5):
  - All algorithms show an initial period of improved generalization
  - RMSE-based methods exhibit overfitting, with DC+RMSE being most severe.
  - NML-based methods see no trend toward overfitting, remaining relatively constant.

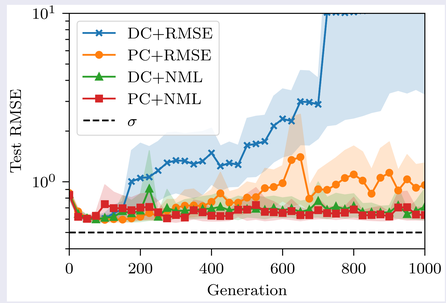

Figure 5: Test fitness during evolution, $\sigma = 0.5$.

- Varying noise (Fig. 6):
  - DC+RMSE has a clear tendency to overfit while PC- and NML-based algorithms have less tendency to overfit
  - For $\sigma = 0$, RMSE algorithms outperform NML algorithms, but, with even a small amount of noise, results suggest that NML-based methods are preferred.
  - PC+NML was found to produce the best generalizability across the noise levels.

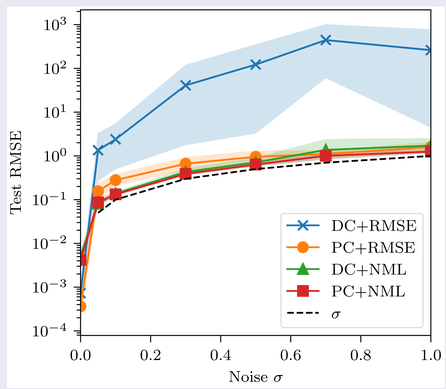

Figure 6: Test fitness for varyinig σ (1000 generations).

Conclusions

Introducing a Bayesian fitness metric was shown to have a substantial impact on the dynamics of GPSR evolution in terms of bloat and overfitting.

- The FBF-based selection algorithm (PC+NML) was shown to be an effective means of regularization which reduced bloat and overfitting.
- Extended evolution resulted not in overfitting or bloat, but rather in slight simplifications in model form (reduced complexity).
- Practical challenges were overcome by combining `SMCPy` and `Bingo`.

References

[1] A. O'Hagan, "Fractional bayes factors for model comparison," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, pp. 99–118, 1995.

[2] S. W. Mahfoud, *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

[3] O. J. Mengshoel and D. E. Goldberg, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," 1999.

Contact: geoffrey.f.bomarito@nasa.gov