ARTEMIS: a Python package for forestry statistics.

Charles Shaw
FixedPoint IO Ltd
August 21, 2023

Abstract

Forestry statistics often grapple with the intricacies of data that might be laden with noise or outliers, making the derivation of accurate growth functions a potentially complex task. ARTEMIS (Advanced Regression and Tree Estimation Model for Integrated Silviculture) has been developed as a solution, offering a package tailored for research and applications in this domain. By harnessing modern optimization techniques, ARTEMIS accurately represents the Chapman-Richards growth function, addressing the inherent challenges of forestry data. Key features include the utilization of evolutionary algorithms for parameter optimization, comprehensive data refinement scripts, and the innovative integration of the Nevergrad optimization library. Notably, ARTEMIS stands out by facilitating the incorporation of prior knowledge into the model, allowing researchers and practitioners to shape the model's search space based on established data, ensuring a well-informed and robust optimization process.

1. Introduction

Forestry, an essential sector that significantly influences the global ecosystem, economy, and climate change mitigation, often relies on accurate statistical models to predict, manage, and understand tree and forest growth. At the heart of this is the need to derive precise growth functions that can encapsulate the nuances of forest dynamics over time. One such widely acknowledged model in forestry statistics is the Chapman-Richards growth function. However, the inherent variability and potential inconsistencies within forestry data, often marked by noise or outliers, challenge the accuracy and reliability of such growth functions.

Traditional techniques, such as regression, can come short when faced with such data sets, leading to the necessity for more robust and adaptable solutions. This gap in the existing methodologies brings forth the relevance and urgency for tools that not only derive accurate growth functions but also possess the flexibility to manage the complexities forestry data presents.

In this paper we introduce ARTEMIS (Advanced Regression and Tree Estimation Model for Integrated Silviculture), a Python package meticulously crafted to address these challenges. It offers a tool for deriving the Chapman-Richards growth function by leveraging modern optimization techniques to the forefront, ensuring that the derived functions are both accurate and reflective of the real-world forestry dynamics.

Beyond the technicalities, there's also the consideration of the vast academic literature and domain knowledge available in forestry. Any modern tool aiming for widespread adoption in this field must facilitate the inclusion of this prior knowledge, ensuring that the derived models aren't just statistically sound but also grounded in decades of research and field observations.

2. ARTEMIS: An Overview of Capabilities and Performance

ARTEMIS is a Python package specifically tailored for forestry statistics. Its primary goal is to deliver precise estimations of tree and forest growth variables. Given the inherent complexities and variabilities present in forestry data, sophisticated techniques are indispensable. ARTEMIS addresses this need by leveraging some of the most advanced algorithms in the domain.

A hallmark of ARTEMIS is its adoption of the Chapman-Richards growth function. This function has consistently demonstrated flexibility and efficacy in modelling tree growth across varied contexts. The integration of this function ensures that ARTEMIS's estimations are anchored in established methodologies.

However, the true differentiator for ARTEMIS is its use of state-of-the-art optimization techniques. Specifically, it uses algorithm an selection wizard, NGOpt, provided by the optimization platform Nevergrad.

NGOpt's design is the result of extensive evaluations of numerous optimizers across a broad spectrum of benchmark tests. Its goal is to use tailored rules to pinpoint the most effective optimizer based on specific problem characteristics. As NGOpt evolves, these rules are fine-tuned, and certain optimizers are swapped out for superior alternatives, enhancing NGOpt's overall performance. This meticulous approach has transformed NGOpt into a sophisticated algorithm selector, allowing it to surpass many prominent standalone optimizers across diverse benchmarks.

2.1. Huber Loss

Huber loss is used for customization of loss function. The Huber loss, often used in robust regression, combines the properties of the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). It behaves similarly to the MSE when errors are small and switches to MAE when errors are large. This characteristic makes the Huber loss less sensitive to outliers compared to the MSE.

Mathematically, the Huber loss is defined as:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \delta \\ \delta\left(|a| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \tag{1}$$

Where:

- $L_\delta(a)$ is the Huber loss.
- a represents the error or residual, i.e., the difference between the predicted and the actual value.
- δ is a threshold that determines the transition point between the MSE-like behaviour and the MAE-like behaviour. For errors smaller than δ, the loss is quadratic, and for larger errors, the loss is linear.

While large errors in MSE have a disproportionately large impact due to the squaring operation, the Huber loss reduces this impact by transitioning to a linear behavior for large errors. This makes the optimization process more stable and less prone to being skewed by extreme data points.

The key advantage of the Huber loss is its robustness to outliers in comparison to the mean squared error (MSE). The function behaves similarly to the mean absolute error (MAE) when the difference between the predicted and actual values is significant (typically for outliers) and mirrors the MSE for smaller differences. This dual behaviour is achieved by the threshold parameter, δ.

- For residuals, the differences between the predicted and actual values, smaller than δ, the Huber loss acts in a quadratic manner, akin to the MSE.
- For residuals surpassing δ, the Huber loss operates linearly, much like the MAE.

This unique combination renders the Huber loss robust against outliers while still maintaining sensitivity towards minor errors.

In essence:

- A **lower Huber loss is preferable** as it signifies a reduced divergence between the predicted and actual values.
- The Huber loss strikes a balance, capturing the sensitivity of the MSE towards minor errors and the resilience of the MAE against outliers.

2.2. NGOpt: Adaptive Algorithm Selection in ARTEMIS

A key component enhancing the capabilities of ARTEMIS is its use of NGOpt, a meta-optimizer within the *Nevergrad* optimization library. 'NGOpt' stands out for its adaptive algorithm selection approach, enabling ARTEMIS to determine the most suitable optimization strategy for the given task.

The forestry domain presents a myriad of optimization challenges, ranging from multi-modal functions to high-dimensional spaces, each requiring distinct algorithmic strategies. Instead of relying on a single optimization method, 'NGOpt' assesses the nature of the problem and dynamically selects the most effective algorithm from its extensive repertoire, which includes but is not limited to Differential Evolution, Genetic Algorithms, and other evolutionary strategies.

This adaptive approach ensures that ARTEMIS remains versatile, effectively tackling a wide array of optimization problems inherent to forestry data. By leveraging 'NGOpt', ARTEMIS optimizes its performance across tasks, providing more accurate and consistent results. The integration of this meta-optimizer exemplifies ARTEMIS's commitment to harnessing state-of-the-art techniques, ensuring that the tool remains at the forefront of forestry analytics.

In the landscape of automatic optimization, challenges emerge from the diverse requirements of real-world problems. These requirements range from the intricacies of problem models to the computational resources at hand. Addressing these challenges requires algorithm selection wizards—tools that are versatile, robust, and adept at selecting the most effective algorithm for a specific problem instance.

The creation of a competitive algorithm selection wizard is a daunting task. It necessitates not only defining the rules for algorithm selection but also configuring the parameters of the selectable algorithms—a challenge in its own right. While automated wizards have been crafted for specific domains like SAT problems, many algorithm selection tools, especially those designed for broader applications, are hand-crafted.

Among these, NGOpt stands out as a paragon of hand-crafted excellence. Integrated within the Nevergrad optimization platform, NGOpt is a product of meticulous research and iterative refinement. Its design was informed by a thorough evaluation of the performance of numerous optimizers across diverse benchmark suites. Based on these insights, hand-crafted rules were devised to strategically select the best optimizer tailored to specific problem features.

The evolution of NGOpt involved iterative enhancements. Over time, certain rules were refined, and specific optimizers were replaced to enhance performance. This rigorous and iterative process birthed a sophisticated algorithm selection wizard. The resultant NGOpt not only embodies the complexity of its design process but also showcases superior performance. When pitted against renowned standalone optimizers across various benchmark suites, NGOpt consistently emerges as a top performer.

For ARTEMIS, the integration of NGOpt means access to a tool that is not just versatile but also empirically validated. By leveraging NGOpt's capabilities, ARTEMIS ensures that its optimization processes are always aligned with the best available strategies, guaranteeing optimal outcomes across diverse forestry challenges.

2.3. Interpreting the 3D Surface Plots for Optimization Landscape

1. Landscape Contours:
    - The plot represents a three-dimensional space where two axes correspond to the model parameters (A and k in this case) and the third axis (vertical) represents the loss (error or objective function value).
    - The 'height' or 'elevation' at any point on this plot represents the error/loss associated with a specific combination of parameters A and k.
    - Low areas or valleys represent parameter combinations with low error, whereas peaks or mountains represent parameter combinations with high error.
2. Optimizer Path:
    - The red line and markers on the plot represent the path taken by the optimization algorithm.

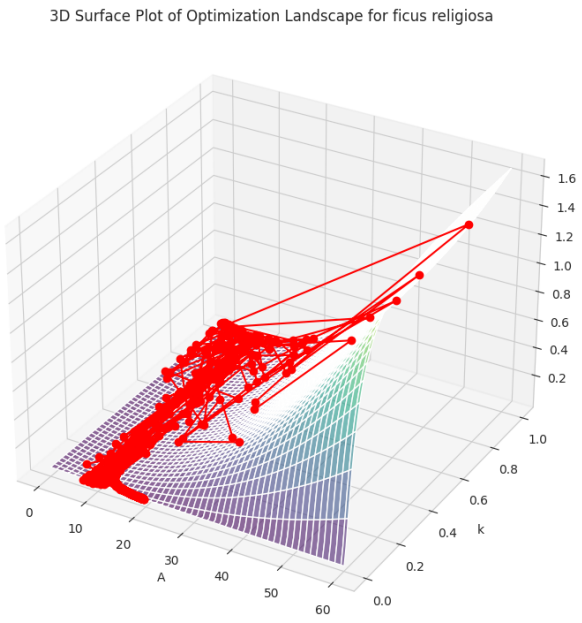

Figure 1: 3D surface plots for optimization landscape:

- Starting from an initial guess, the optimizer tries different combinations of parameters in its quest to find the lowest point on the landscape (the best parameters).
- By following the path, you can see how the optimizer 'searched' through the parameter space.

3. Global vs Local Minima:
   - A critical feature to look out for in such plots is the presence of multiple low areas (valleys). This indicates that there may be multiple sets of parameters that result in a similar low error.
   - The true goal of the optimizer is to find the global minimum (the absolute lowest point across the entire landscape). However, sometimes optimizers can get stuck in local minima (a point lower than its surroundings, but not the lowest overall).
   - If you see multiple valleys and the optimizer's path ends in one of them, it might be stuck in a local minimum. It's essential to know this as there might be better parameter sets the optimizer did not explore.
4. Initial Guess and Convergence:
   - The starting point of the optimizer path indicates the initial guess for the parameters.
   - The end of the path represents the best parameters found by the optimizer.
   - If the path shows a clear and steady descent into a valley, it indicates good convergence of the optimization algorithm. If the path seems erratic or doesn't settle into a valley, it might suggest challenges with optimization convergence.
5. Complexity of Landscape:
   - A smooth landscape with a clear single valley is typically easier for optimization algorithms. A rugged landscape with multiple valleys and peaks suggests a more complex optimization problem, which may require more sophisticated or randomized optimization techniques.

3. Data Handling and Pre-processing

Forestry data, inherently complex and often gathered from varied sources, requires meticulous handling and preprocessing. ARTEMIS addresses this need head-on. Its advanced data cleaning scripts, tailored specifically for forestry datasets, tackle a plethora of common issues. These scripts systematically identify and rectify missing values, spurious outliers, and inconsistent data entries. Beyond mere cleaning, ARTEMIS also offers features for data transformation, ensuring that the data conforms to the requirements of the subsequent analytical steps.

The significance of this rigorous preprocessing cannot be overstated. In the realm of forestry, even minor data inconsistencies can lead to substantial discrepancies in predictions. By ensuring that the input data is of the highest quality, ARTEMIS substantially enhances the reliability of its analytics, laying a strong foundation for all subsequent analyses.

This code first winsorizes (at 1%) outliers for "height", then fits the Chapman-Richards growth function to the winsorized data and plots the results in a multi-subplot figure, saving the plot as a PNG file in the specified directory.

4. Handling of Outliers

In the domain of forestry analytics, the treatment of outliers requires careful and nuanced consideration. Unlike certain other fields where outliers can straightforwardly be deemed as anomalies, forestry presents a unique challenge. Genuine biological variability can sometimes manifest as extreme values in the dataset, reflecting rare but natural occurrences. For instance, a tree of a particular species might exhibit an unusually tall height due to specific genetic or environmental factors. On the other hand, spurious outliers could arise from various sources, such as measurement errors, data entry mistakes, or inconsistencies in data collection methodologies.

Given this backdrop, we have taken the decision to apply Winsorising. Winsorising, in particular, involves limiting the extreme values of a dataset, either by capping them at a predetermined threshold or replacing them with more central values. In the context of our application:

```
DATA['height'] = mstats.winsorize(DATA['height'], limits=[0, 0.001])
DATA['dbh'] = mstats.winsorize(DATA['dbh'], limits=[0, 0.001])
```

The above lines of code depict the application of Winsorisation to the height and dbh (Diameter at Breast Height) columns of the dataset. By default, we Winsorise at the 99.9th percentile. This approach primarily aims to mitigate the influence of extreme outliers that might stem from inadvertent factors, such as logging errors.

While Winsorisation offers one layer of robustness, our methodology integrates further strategies to ensure model resilience in the face of outliers. Specifically, we employ the Huber loss function during the model training phase. Unlike traditional loss functions, the Huber loss is particularly tailored to reduce the influence of outliers. For residuals below a certain threshold, it operates akin to the Mean Squared Error, while for larger residuals, its behaviour aligns more with the Mean Absolute Error. This dual nature ensures that our model doesn't excessively bend to accommodate extreme data points.

In tandem with the Huber loss, our approach harnesses the power of the Nevergrad optimisation library, leveraging its evolutionary-type search algorithms. Traditional gradient-based optimisation methods can be perturbed by outliers, leading to sub-optimal model parameters. In contrast, Nevergrad's derivative-free optimisation, inspired by principles of biological evolution, offers inherent robustness against outliers.

In conclusion, our multifaceted strategy, which encompasses Winsorisation, the Huber loss, and Nevergrad's evolutionary algorithms, aims to strike a delicate balance. While we recognise the need to account for genuine biological variability, we simultaneously strive to insulate our model from the undue influence of spurious outliers.

5. Modelling and Growth Functions

Central to ARTEMIS's analytical engine is the Chapman-Richards growth function. This mathematical expression, grounded in empirical research, has consistently demonstrated its ability to capture the intricacies of tree growth across diverse contexts. The function delineates the sigmoidal growth curve commonly observed in trees, encapsulating the nuances of the rapid initial growth which gradually tapers as trees approach maturity.

While various growth functions exist in silviology, the Chapman-Richards function's empirical accuracy and theoretical soundness make it the cornerstone of ARTEMIS's modelling strategy. By anchoring its predictions in this well-established function, ARTEMIS ensures that its outputs resonate with both empirical observations and theoretical expectations.

6. Chapman-Richards growth function

Growth functions in general describe the change in size of an individual or population with time (Burkhart and Tome, 2012). The Chapman-Richards growth function can be described as

$$y(t) = y_{\max}(1 - e^{-kt})^p \tag{2}$$

Assume that $y(t)$ is a tree growth variable, in our case tree dbh, and $y_{\max}$ is the maximum value this growth variable can take (in absolute terms for a given species in general or for a given species on a given site) then the term $[1–e^{-kt}]^p$ is a modifier reducing the maximum growth variable to its current state at time t. k is an empirical growth parameter scaling the absolute growth rate. The empirical parameter p is related to catabolism (destructive metabolism), which is said to be proportional to an organism's mass. It is often restricted to a value of three or four for theoretical, biological reasons.

7. Optimization Techniques

In the face of complex and often noisy forestry data, mere modelling is insufficient. The models need to be fine-tuned to the data, a task that demands sophisticated optimization techniques. ARTEMIS, understanding this imperative, integrates the Nevergrad library—a cutting-edge optimization toolbox. Within Nevergrad, ARTEMIS predominantly employs NGOpt, a versatile meta-optimizer.

NGOpt stands out for its adaptive algorithm selection. Instead of being confined to a single optimization strategy, NGOpt assesses the problem at hand and judiciously selects the most effective algorithm, be it Differential Evolution, Genetic Algorithms, or other advanced techniques. This adaptability ensures that ARTEMIS's optimization process is always attuned to the specific challenges of the dataset, yielding optimal or near-optimal solutions consistently.

8. Interactive Features and User Experience

Beyond its analytical capabilities, ARTEMIS prioritises user engagement and experience. The software's design integrates an interactive species panel, a feature that revolutionises data exploration. This panel offers real-time, species-specific data visualisation, allowing users to glean insights with unprecedented granularity. Such interactive features transform the user's engagement from passive observation to active exploration, fostering a deeper understanding of the data.

Furthermore, ARTEMIS's intuitive interface ensures that its advanced capabilities are accessible to a non-technical user. The streamlined Jupyter Workbook, coupled with comprehensive documentation, ensures that users, irrespective of their technical proficiency, can navigate the software, harness its features, and extract meaningful insights with ease.

References

[1] Burkhart, H. and Tome, M., 2012. Modeling forest trees and stands. Springer