Generating Rational Elementary Integrable Expressions

Rashid Barket

Coventry University

Abstract

We propose a method to generate mathematical expressions that are guaranteed to be elementary integrable. The method takes advantage of the Risch algorithm and specifically the Trager-Rothstein method. Using these two ideas, we are able to generate elementary integrable expressions in an algebraically closed field.

1 Introduction

One of the main factors for a successful machine learning model is having the right type of data. In the field of mathematics, gathering data is a relatively "cheap" task (no need for sensors, cameras, etc.). However, that does not make the task of data generation trivial. One such area where data generation is non-trivial is to generate mathematical expressions that can be integrated. Generating expressions is a relatively simple task, but the specific goal is to generate elementary expressions that when integrated, also produce an elementary expression. When generating our data, there are a couple goals to keep in mind: The dataset should be (at least partially) representative of actual inputs that Computer Algebra System users would enter and the expressions should represent many different classes of functions (e.g. elementary, complex, or "special" functions like $\int \frac{1}{x} = \mathrm{li}(x)$).

2 Related Work

Currently, there are not that many (public) benchmark datasets in the world of Computer Algebra for integration. Literature that talk about integration in Computer Algebra focus more on the theoretical aspect of the computational complexity and do not have proper benchmarks, for e.g. [1, 3]. In 2020, Lample & Charton used a transformer (a machine learning architecture) to predict an integral for a given integrand [2]. An algorithm was developed to generate expressions of varying length, operators, and operands. It then remained to find the integral given an expression. Three methods were developed:

- FWD: Integrate an expression f through a Computer Algebra System (CAS) to get F and add the pair (f, F) to the dataset.
- BWD: Differentiate an expression f to get f' and add the pair (f', f) to the dataset.
- IBP: Given two expressions f and g, calculate f' and g'. If $\int f'g$ is known then the following holds (integration-by-parts):

These methods are good to a certain extent. There are biases in the types of expressions made, mainly around the length of the expressions. An example is that in the FWD method, the integrand tends to, on average, be longer than the integral, meaning that the expressions in the data set would all have this additional property if we didn't include the other two methods. A more detailed critique by Piotrowski et al. [4] shows the weaknesses of these data generation methods. It would be preferable to generate data that samples *all* types of elementary integrable expressions, to avoid overfitting on certain types of expressions and overall produce a more robust model.

3 The Risch Algorithm

The Risch algorithm [5] is a procedure for determining whether an elementary function has an elementary indefinite integral and if it does, to compute the integral. This algorithm is a key feature of all CASs as part of the integration function. However, no CAS has fully implemented the entire algorithm as it is very large (100+ pages) with many special cases.

Rather than using the Risch algorithm to determine whether an expression is elementary integrable, we will instead use the Risch algorithm to generate integrable expressions for our dataset. The basic premise of the Risch Algorithm is that given an expression $f(x) = a(x)/b(x)$, it will break the expression into its *polynomial part* and its *rational part* i.e.

$$\int a(x)/b(x) = \int P(x) + \int R(x)/b(x)$$

where $P(x)$ is the polynomial part and $R(x)/b(x)$ is the rational part. Our focus will remain on generating rational elementary integrable expressions as it is a tougher problem than the polynomial case.

The key to generating rational elementary integrable expressions is through one of the key algorithms for dealing with this integral: The Trager-Rothstein method. A theorem in this method tells is that $\frac{R(x)}{b(x)}$ is elementary integrable if and only if the roots of the Trager-Rothstein resultant, $\text{res}(R(x) - zb'(x), b(x)$, is a constant. We take advantage of this property and learn of ways to generate expressions that will have a constant Trager-Rothstein resultant.

Let us demonstrate this method with an example: the function $f(x) = \frac{a(x)}{log(x)+3}$. When we calculate the TR-resultant $\text{res}\big(a(x) - z(\frac{1}{x}), log(x) + 3\big)$, we find that the root of the Trager-Rothstein resultant is $a(x)x$. Since the root has to be a constant, this means $a(x)x = C$ for a constant $C \in \mathbb{Q} \implies a(x) = C/x$. Thus, we have found all the possible numerators $a(x)$ that make $f(x)$ elementary integrable. This is the basic premise of this method to generate elementary integrable expressions and is expanded upon much more in my current research.

References

[1] James H. Davenport. The parallel Risch algorithm (i). In *Computer Algebra*, pages 144–157, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[2] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *Proc. International Conference on Learning Representations 2020*, 2020.

[3] Emmanuel Manuel. Integration of elementary functions. *Journal of Symbolic Computation*, 9(2):117–173, 1990.

[4] Bartosz Piotrowski, Josef Urban, Chad E. Brown, and Cezary Kaliszyk. Can neural networks learn symbolic rewriting? In *Proc. Artificial Intelligence and Theorem Proving*, 2019.

[5] Robert H Risch. The problem of integration in finite terms. *Transactions of the American Mathematical Society*, 139:167–189, 1969.