Mathematical Modeling and Analysis

Numerical differentiation of noisy data

Rick Chartrand, rickc@lanl.gov

It is very common in scientific applications to have to compute the derivative of a function specified by data. Particularly in the case of experimental data, substantial noise or imprecision may be present. Straightforward techniques for numerically calculating derivatives will amplify this noise, often so much that the result is useless. We propose a new algorithm that can give accurate results even in the presence of a large amount of noise. This approach also has the benefit of not forcing the result to be continuous, which allows for the detection of corners or edges in the data.

The most common approach to computing derivatives is finite differencing. At its simplest, this is just subtracting each data point from the one following it, and dividing by the difference between them. So, for a function f defined at data points $0 = x_1, x_2, \dots, x_N = L$, one computes

$$f'(x_i) \sim \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \qquad (1)$$

This is just the slope of the line joining two adjacent data points. The name "finite differencing" distinguishes this from "infinitesimal differencing:" the value of $f'(x_i)$ can be thought of as the right-side of (1) for x_{i+1} "infinitesimally close" to x_i. Since we only have finitely many data points, (1) is about as good as we can do.

To see why this fails for noisy data, look at the function in Figure 1(a). Although there are trends in the data that are clear to the eye, slopes of adjacent points fluctuate wildly, as in Figure 1(b).

If the problem is the noise in the data, why not denoise the data first? There are many denoising algorithms. None of them are perfect. This means that some noise or inaccuracy can be expected to remain in the data. The differentiation process will still amplify this noise, giving an unsatisfactory result. See Figure 2, in which total-variation regularization (see below) is used to denoise the image before computing the derivative with finite differencing.

The alternative we propose is to *regularize* the differentiation process itself. We constrain the possible outcomes in a way which guarantees that the resulting derivative will not be noisy. The way we do this is by requiring that the derivative u be the minimizer of the following functional:

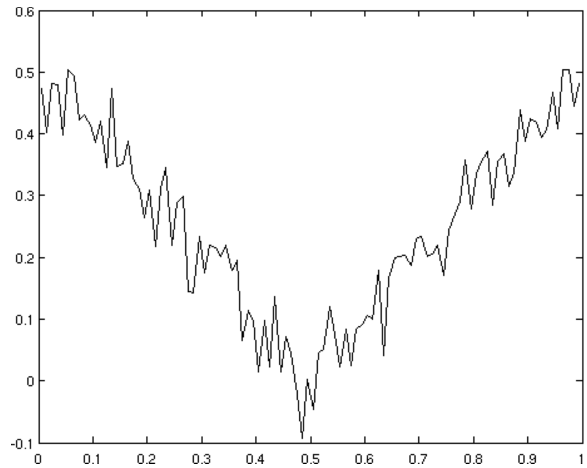
$$F(u) = \alpha \int_0^L |u'| + \int_0^L |Au - f|^2. \qquad (2)$$

Here, A is the operator of antidifferentiation: $Au(x) = \int_0^x u$. The purpose of the second term of (2) is to make sure that what we are computing is consistent with the function we are given: the antiderivative of u should not stray too far from f. The first term is the *total variation* of u, which measures the total of all the ups and downs in u. Keeping this term small ensures that the result will not be noisy, as noise has a very large total variation. The parameter α controls the relative importance of the two terms. If α is chosen correctly, the result will be a derivative u that is as "regular" as possible, while having Au be within the noise level of f.

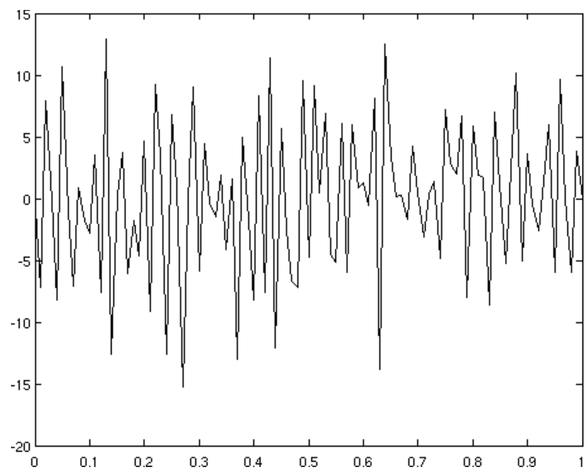
Total-variation regularization has been used in other contexts. An example is image denoising, where a two-dimensional analog of (2) is used, but with Au replaced by u [1].

Other types of regularizations have been applied to the differentiation process. The first was proposed by Cullum [2]. In her approach, $\int |u'|^2$ is used in place of the total variation term in (2). The effect of this regularization, as well as others proposed since, is to force the resulting u to be continuous. This is because $\int |u'|^2$ is infinite if u is discontinuous. On the other hand, the total variation of u is unaffected by discontinuities. It only measures changes in u, whether sharp or gradual. Thus, total-variation regularization has the advantage of being able to accurately compute derivatives that are discontinuous.

Figure 3 shows the result of using total-variation regularization to differentiate the function in Figure 1(a). The general shape is captured almost perfectly: two constant portions with



(a)



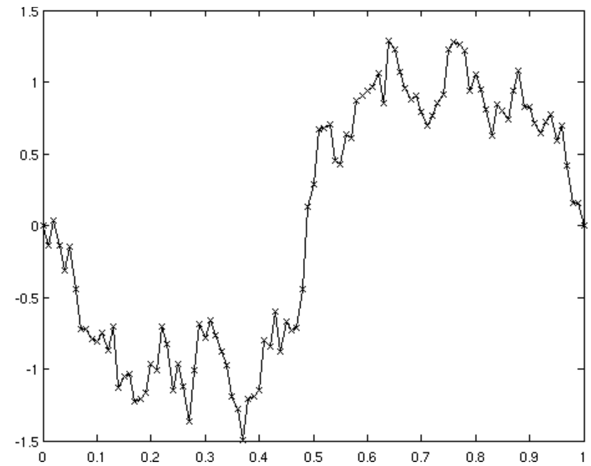
(b)

A noisy function (a) and its derivative (b) computed by finite differencing. The noise is amplified to the point of uselessness.

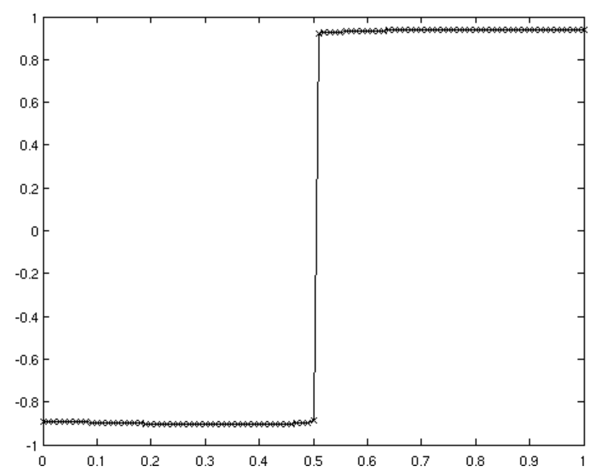
a sharp jump in the middle. Ideally, the values would be ± 1; the jump size is a little too small, as that reduces the total variation. Methods exist for correcting such artifacts; implementing them is the subject of current work.

Acknowledgements

Los Alamos Report LA-UR-05-9309. Funded by the Department of Energy under contracts W-7405-ENG-36 and the DOE Office of Science's Advanced Scientific Computing Research (ASCR) program in Applied Mathematical Sciences.



The function in Figure 1(a) is denoised before differentiating. The result is still noisy and inaccurate.



The derivative of the function in Figure 1(a), computed using total-variation regularization. The resulting noiseless derivative has a sharp jump in the correct location.

References

[1] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[2] J. Cullum. Numerical differentiation and regularization. *SIAM J. Numer. Anal.*, 34:1779–1791, 1971.