AMATH 563: APPROXIMATING THE LEVY FUNCTION IN 1D

WINNIE LAU

Department of Applied Mathematics, University of Washington, Seattle, WA
wwlau@uw.edu

1. Introduction

Given a Levy function in 1D and a finite number of observed points with some noise, the goal of this report is to approximate the original function using only the limited and noisy observations. A model is introduced as a linear combination of a collection/dictionary of functions where a least squares model objective function was used to identify the best fit coefficients. This report uses the dictionaries for the cosine series, monomials, and the Radial Basis Function (RBF) with various values of N and λ. Further investigation was done with the sine series, sum of cosine and sine series, and the logistic sigmoid basis functions. The original function was most closely fit with the RBF and logistic sigmoid basis given a larger N and a smaller λ.

2. Methods

In this report, the function we want to estimate is the Levy function in 1D defined as

$$f(x) = \sin(\pi w)^2 + (w-1)^2\left(1+\sin(2\pi w)^2\right), \quad w = 1 + \frac{10x-1}{4}, \quad x \in [-1,1], \tag{1}$$

We then define uniformly space points $\{x_k\}_{k=1}^{K} \in [-1,1]$ with $K = 50$. Sampling (1) at these points with added noise ξ_k yields the observations

$$y_k = f(x_k) + \xi_k, \qquad \xi_k \sim N(0,\sigma^2), \quad k = 0, \dots K-1.$$

where we have some standard deviation $\sigma > 0$ for the noise. Let $\mathbf{y} \in \mathbb{R}^K$ be the vector containing the values of y_k. We introduce a model

$$g(x) = \sum_{n=0}^{N-1} c_n \psi_n(x),$$

for some integer $N \geq 1$, coefficients $c_n \in \mathbb{R}$, and a dictionary of functions ψ_n. To approximate (1) over the dictionary $\{\psi_n\}_{n=0}^{N-1}$, we solve the optimization problem for the best fit coefficients

$$\widehat{c} := \arg\min_{\boldsymbol{c}} \frac{1}{2}\sum_{k=0}^{K-1} |g(x_k) - y_k|^2 + \frac{\lambda}{2}\|\boldsymbol{c}\|_2^2, \quad \text{subject to} \quad g(x) = \sum_{n=0}^{N-1} c_n \psi_n(x). \tag{2}$$

where $\mathbf{c} \in \mathbb{R}^N$ is the vector of c_n coefficients and $\lambda > 0$ is a regularization parameter. To rewrite this as a least squares problem, we plug $g(x_k)$ into (2) to get

Date: April 14, 2025.

$$\begin{aligned}\widehat{c} &= \arg\min_{\boldsymbol{c}} \frac{1}{2}\sum_{k=0}^{K-1}\left|\sum_{n=0}^{N-1} c_n\psi_n(x_k) - y_k\right|^2 + \frac{\lambda}{2}\|\boldsymbol{c}\|_2^2 \\ &= \arg\min_{\boldsymbol{c}\in\mathbb{R}^N} \frac{1}{2}\left\|\Psi\mathbf{c} - \mathbf{y}\right\|^2 + \frac{\lambda}{2}\|\boldsymbol{c}\|_2^2\end{aligned}$$

for a matrix $\Psi \in \mathbb{R}^{K\times N}$ defined as

$$\Psi = \begin{bmatrix} \psi_0(x_0) & \psi_1(x_0) & \dots & \psi_{N-1}(x_0) \\ \psi_0(x_1) & \psi_1(x_1) & \dots & \psi_{N-1}(x_1) \\ \vdots & & \ddots & \vdots \\ \psi_0(x_{K-1}) & & \dots & \psi_{N-1}(x_K - 1) \end{bmatrix}.$$

Taking the gradient and finding the critical point to solve for $\widehat{c}$ yields

$$\begin{aligned}\nabla\left(\frac{1}{2}\left\|\Psi\mathbf{c} - \mathbf{y}\right\|^2 + \frac{\lambda}{2}\|\boldsymbol{c}\|_2^2\right) &= 0 \\ \Psi^T(\Psi\widehat{\mathbf{c}} - \mathbf{y}) + \lambda\widehat{\mathbf{c}} &= 0 \\ (\Psi^T\Psi + \lambda I)\widehat{\mathbf{c}} &= \Psi^T\mathbf{y} \\ \widehat{\mathbf{c}} &= (\Psi^T\Psi + \lambda I)^{-1}\Psi^T\mathbf{y}\end{aligned}$$

We now have our optimal model

$$\widehat{f} = \sum_{n=0}^{N-1} \widehat{c}_n\psi_n,$$

which we will compute for the dictionaries

- (Cosine series) $\psi_n(x) = \cos(n\pi x)$
- (Monomials) $\psi_n(x) = x^n$
- (RBF basis) $\psi_n(x) = \exp\left(-\frac{\|x-z_n\|^2}{2\ell^2}\right)$ for $\ell = 0.2$ and $z_n = -1 + 2n/N$.
- (Sine series) $\psi_n(x) = \sin(n\pi x)$
- (Cosine & Sine series) $\psi_n(x) = \cos(n\pi x) + \sin(n\pi x)$
- (Logistic Sigmoid) $\psi_n(x) = \frac{1}{1+e^{k(x-z_n)}}$ for $k = 10$, z_n as defined previously

for $N = 10, 50, 100$ and $\lambda = 10^{-10}, 10^{-2}, 1$. The last three dictionaries were chosen in order to investigate the differences between the cosine series and the sine series, along with their sum which is similar to the Fourier series, except the cosine and sine now have the same coefficient for some n. This was implemented in Python where $\widehat{c}$ was found using the function `np.linalg.solve()`.

3. Results

The function $\widehat{f}$ was plotted with respect to the different basis functions and N values, and λ was compared within each graph, as seen in Figure 1.

For the cosine basis, we see that as N increased, there are increased oscillations in the resulting plot.The amplitude also increased as x moved further away from 0 inn order to fit the increasing $f(x_k)$. Changing the values of λ for each N also led to little difference as the graphs mainly overlapped each other.

For the monomial basis, we see the closest estimate for $f(x)$ at $N = 10$. We can also note that the graph for $\lambda = 1$ is smoother than the graph for $\lambda = 0.01$ as the orange line oscillates above and below the green one. As N increased, the tails became very large and inaccurate for $\lambda = 1e - 10$.

We see the best fit occurs with the RBF function when $N = 100$, $\lambda = 1e - 10$ where it most closely following the oscillating behavior of the Levy function. As λ increased, the approximations

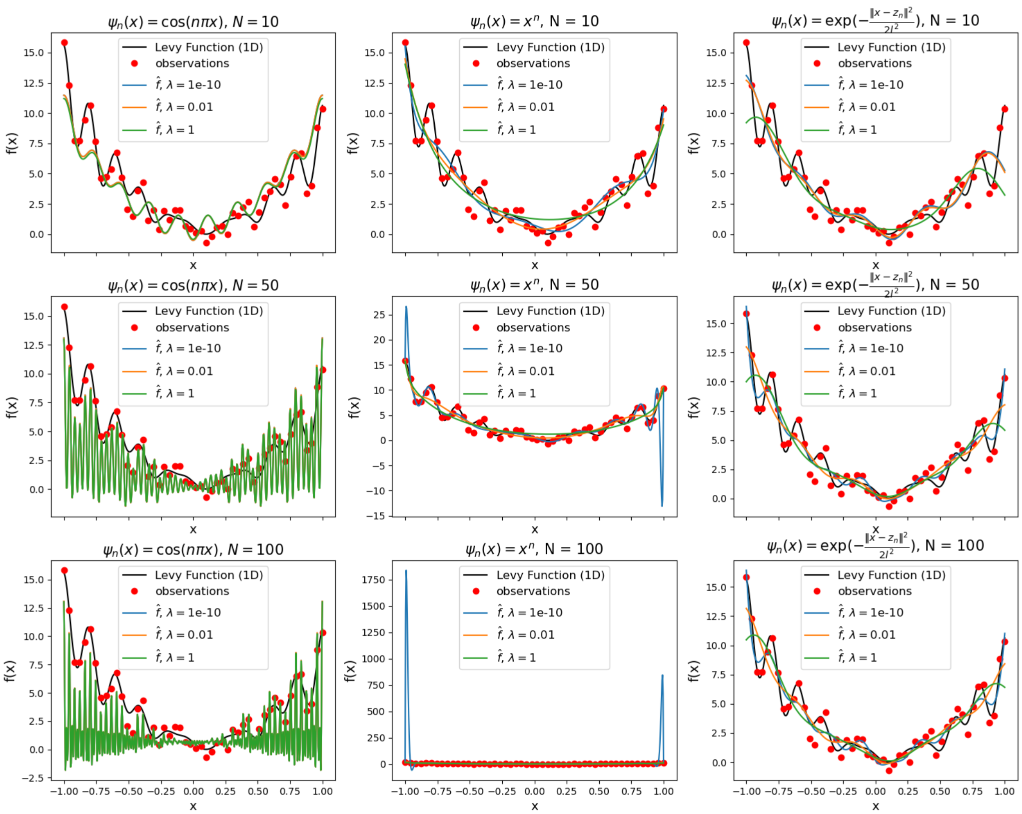

FIGURE 1. Graphs of $\widehat{f}$ with each column being the dictionaries cosine basis, monomial basis, and RBF, while each row is $N = 10, 50, 100$. Each plot contains the Levy function, the observations, and the approximations with $\lambda = 1e - 10, 1e - 2, 1$.

were not as exact, particularly on the tails, but the models were still able to follow the general curve of the function.

Three additional bases were considered for this report: the sine basis, the sum of cosine and sine, and the logistic sigmoid basis (Figure 2). For the sine basis, this did the worse in fitting the observations as it could not interpolate $f(x)$ around the tails. We also see the number of oscillations going up as N increased, similar to the cosine basis. Meanwhile, for the basis containing both cosine and sine, we see this was able follow $f(x)$ and interpolate the observations as N increased. The graphs for all values of λ also behaved similarly as they mostly overlapped each other.

For the logistic sigmoid basis, we see that the $\widehat{f}$ matched the Levy function most accurately for $N = 50, 100$ and $\lambda = 1e - 10$. As λ increased, the resulting $\widehat{f}$ became smoother, while as N increased, $\widehat{f}$ matched $f(x)$ more closely.

4. Discussion and Conclusions

From our results, we see that as N increased, the approximation $\widehat{f}$ was able to interpolate the observations much more closely. Meanwhile, the opposite was true for λ, mainly with the RBF and logistic sigmoid basis, where as λ decreased, it matched the original function much more

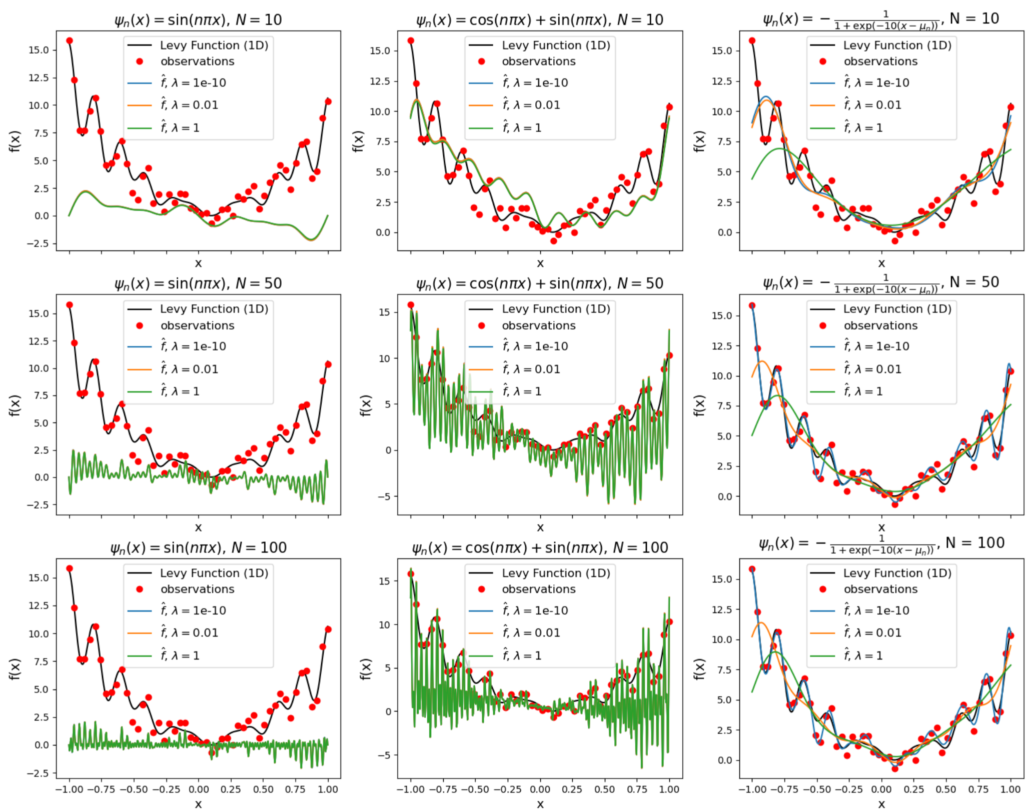

FIGURE 2. Three additional dictionaries were tested: the sine basis, sum of cosine and sine basis, and the logistic sigmoid basis. Each row represents $N = 10, 50, 100$ respectively, and each plot contains the Levy functions, the observations, and approximations with $\lambda = 1e-10, 1e-2, 1$.

closely. We can also note that as λ increased, it smoothed out the approximation, though it had an adverse effect for the monomial basis as the tails of the approximation blew up with a small λ. For the periodic bases, λ had less of an effect. This illustrates how we may end up with overfitting, particularly with an overdetermined system with a large N and small λ that closely interpolates the given observations but may fail for any new points given.

Future directions for this report includes exploring other values of N and λ, along with their effects on other dictionaries to understand their effects and how well we can create an approximation for the Levy function.

ACKNOWLEDGEMENTS

The author is thankful to Prof. Bamdad Hosseini for useful discussions about the implementation of the model in this report. We are also thankful to David Kieke, Emily Zhang, Christina Nicolaides, and Howard Cheng for discussions in understanding the theory and implementing the functions in Python.