Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2021-0062

Vol. **14**, No. 4, pp. 839-861
November 2021

Nonlinear Level Set Learning for Function Approximation on Sparse Data with Applications to Parametric Differential Equations

Anthony Gruber[1,*], Max Gunzburger[1], Lili Ju[2], Yuankai Teng[2] and Zhu Wang[2]

[1] *Department of Scientific Computing, Florida State University, 400 Dirac Science Library, Tallahassee, FL 32306, USA*
[2] *Department of Mathematics, University of South Carolina, 1523 Greene Street, Columbia, SC 29208, USA*

Received 23 April 2021; Accepted (in revised version) 6 August 2021

Abstract. A dimension reduction method based on the "Nonlinear Level set Learning" (NLL) approach is presented for the pointwise prediction of functions which have been sparsely sampled. Leveraging geometric information provided by the Implicit Function Theorem, the proposed algorithm effectively reduces the input dimension to the theoretical lower bound with minor accuracy loss, providing a one-dimensional representation of the function which can be used for regression and sensitivity analysis. Experiments and applications are presented which compare this modified NLL with the original NLL and the Active Subspaces (AS) method. While accommodating sparse input data, the proposed algorithm is shown to train quickly and provide a much more accurate and informative reduction than either AS or the original NLL on two example functions with high-dimensional domains, as well as two state-dependent quantities depending on the solutions to parametric differential equations.

AMS subject classifications: 65D15, 65D40

Key words: Nonlinear level set learning, function approximation, sparse data, nonlinear dimensionality reduction.

1. Introduction

It is frequently the case that scientists or engineers need to draw conclusions about the output of a function based on limited or incomplete data. Such situations arise, for example, when the output depends on the solution of expensive differential equations,

[*] Corresponding author. *Email addresses:* agruber@fsu.edu (A. Gruber), mgunzburger@fsu.edu (M. Gunzburger), ju@math.sc.edu (L. Ju), yteng@email.sc.edu (Y. Teng), wangzhu@math.sc.edu (Z. Wang)

or when lack of time and resources precludes the collection of sufficient high-quality samples. When this occurs, it becomes critical to maximize the value of the limited resources at hand, which requires informed algorithms for dimension reduction.

More specifically, let $U \subset \mathbb{R}^n$ be a bounded domain and consider the problem of approximating a continuously differentiable scalar function $f : U \to \mathbb{R}$ based on some predefined samples $\{\mathbf{x}^s, f(\mathbf{x}^s), \nabla f(\mathbf{x}^s)\}_{s\in S}$ of the function and its gradient vector field. Note that f may represent either a scalar quantity or some component of a vector quantity, so that no generality is lost with this consideration. Additionally, let $\rho : \mathbb{R}^n \to \mathbb{R}^+$ be a probability density function supported on U such that f is square-integrable with respect to ρ, i.e.

$$\|f\|_2^2 := \int_U f(\mathbf{x})^2 \rho(\mathbf{x})\, dx < \infty.$$

To generate a pointwise approximation to f, it is reasonable to seek a function $\tilde{f} : U \to \mathbb{R}$ which satisfies the minimization condition

$$\tilde{f}(\mathbf{x}) \in \operatorname*{arg\,min}_{g\in C^1(U)} \left\|f(\mathbf{x}) - g(\mathbf{x})\right\|_2^2. \tag{1.1}$$

However, if the dimension n is large relative to the number $|S|$ of available samples (i.e. sparse data), training a regression model to approximate f directly becomes infeasible. Indeed, unless the training data itself has a hidden low-dimensional structure, unsupervised learning methods such as feed-forward neural networks are prone to overfitting, leading to poor accuracy on new data as a result of inadequate generalizability. Therefore, it is necessary to employ some kind of dimension reduction to increase the density of the sampling data to the point where it is useful for approximating solutions to (1.1).

A prototypical example of this issue arises when studying the numerical solutions of differential equations with limited computational budget. Let I be a multi-index, $\boldsymbol{\beta} \in \mathbb{R}^m$, and consider a k-th order parameterized system of $R \in \mathbb{N}$ partial differential equations (PDE) for the function $\mathbf{u} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^l$,

$$F^r\left(\boldsymbol{\beta}, \mathbf{x}, \mathbf{u}, \frac{\partial^{|I|}\mathbf{u}}{\partial x^I}\right) = 0, \quad 1 \le r \le R, \quad 1 \le |I| \le k, \tag{1.2}$$

which may depend on some number of initial or boundary conditions. Suppose additionally that the assignment $\boldsymbol{\beta} \mapsto \mathbf{u}(\boldsymbol{\beta}, \mathbf{x})$ is unique, so that solutions to (1.2) are parameterized by the variables $\boldsymbol{\beta}$. For prediction and sensitivity analysis it is often necessary to compute the value of some functional $\mathcal{K}(\mathbf{u})$ on PDE solutions $\mathbf{u} \in C^k(\mathbb{R}^m \times \mathbb{R}^n; \mathbb{R}^l)$ (e.g. temperature or total kinetic energy) which is implicitly a function of the parameters $\boldsymbol{\beta}$, i.e. $\mathcal{K}(\boldsymbol{\beta}) = \mathcal{K}(\mathbf{u}(\boldsymbol{\beta}, \mathbf{x}))$. On the other hand, it is usually not feasible to simulate the (potentially expensive) system (1.2) for every parameter configuration desired, so it is necessary to have a reasonable yet inexpensive approximation to $\mathcal{K}$ which can be computed for any $\boldsymbol{\beta}$ in place of numerically solving (1.2). In the language of before, this means finding $\tilde{\mathcal{K}}$ satisfying

$$\tilde{\mathcal{K}}(\boldsymbol{\beta}) \in \operatorname*{arg\,min}_{\mathcal{G}:\mathbb{R}^m\to\mathbb{R}} \left\|\mathcal{K}(\boldsymbol{\beta}) - \mathcal{G}(\boldsymbol{\beta})\right\|_2^2.$$