Appl. Comput. Harmon. Anal. 37 (2014) 162–170

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

Inverting nonlinear dimensionality reduction with scale-free radial basis function interpolation

Nathan D. Monnig [a,*], Bengt Fornberg [a], François G. Meyer [b]

[a] Department of Applied Mathematics, UCB 526, University of Colorado at Boulder, Boulder, CO 80309, United States
[b] Department of Electrical Engineering, UCB 425, University of Colorado at Boulder, Boulder, CO 80309, United States

ARTICLE INFO

Article history:
Received 1 May 2013
Received in revised form 24 September 2013
Accepted 29 October 2013
Available online 8 November 2013
Communicated by Charles K. Chui

Keywords:
Inverse map
Nonlinear dimensionality reduction
Radial basis function
Interpolation
Nyström extension

ABSTRACT

Nonlinear dimensionality reduction embeddings computed from datasets do not provide a mechanism to compute the inverse map. In this paper, we address the problem of computing a stable inverse map to such a general bi-Lipschitz map. Our approach relies on radial basis functions (RBFs) to interpolate the inverse map everywhere on the low-dimensional image of the forward map. We demonstrate that the scale-free cubic RBF kernel performs better than the Gaussian kernel: it does not suffer from ill-conditioning, and does not require the choice of a scale. The proposed construction is shown to be similar to the Nyström extension of the eigenvectors of the symmetric normalized graph Laplacian matrix. Based on this observation, we provide a new interpretation of the Nyström extension with suggestions for improvement.

1. Introduction

The construction of parametrizations of low dimensional data in high dimension is an area of intense research (e.g., [1–4]). A major limitation of these methods is that they are only defined on a discrete set of data. As a result, the inverse mapping is also only defined on the data. There are well known strategies to extend the forward map to new points—for example, the Nyström extension is a common approach to solve this *out-of-sample extension* problem (see e.g., [5] and references therein). However, the problem of extending the inverse map (i.e. the *preimage problem*) has received little attention so far (but see [6]). The nature of the preimage problem precludes application of the Nyström extension, since it does not involve extension of eigenvectors.

We present a method to numerically invert a general smooth bi-Lipschitz nonlinear dimensionality reduction mapping over all points in the image of the forward map. The method relies on interpolation via radial basis functions of the coordinate functions that parametrize the manifold in high dimension.

The contributions of this paper are twofold. Primarily, this paper addresses a fundamental problem for the analysis of datasets: given the construction of an adaptive parametrization of the data in terms of a small number of coordinates, how does one synthesize new data using new values of the coordinates? We provide a simple and elegant solution to solve the "preimage problem". Our approach is scale-free and numerically stable and can be applied to any nonlinear dimension reduction technique. The second contribution is a novel interpretation of the Nyström extension as a properly rescaled

* Corresponding author.
E-mail address: nathan.monnig@colorado.edu (N.D. Monnig).
URL: http://amath.colorado.edu/people/nathan-monnig (N.D. Monnig).

1063-5203/$ – see front matter
http://dx.doi.org/10.1016/j.acha.2013.10.004

radial basis function interpolant. A precise analysis of this similarity yields a critique of the Nyström extension, as well as suggestions for improvement.

2. The inverse mapping

2.1. Definition of the problem, and approach

We consider a finite set of n datapoints $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\} \subset \mathbb{R}^D$ that lie on a bounded low-dimensional smooth manifold $\mathcal{M} \subset \mathbb{R}^D$, and we assume that a nonlinear mapping has been defined for each point $\boldsymbol{x}^{(i)}$,

$$\boldsymbol{\Phi}_n : \mathcal{M} \subset \mathbb{R}^D \longrightarrow \mathbb{R}^d \tag{1}$$

$$\boldsymbol{x}^{(i)} \longmapsto \boldsymbol{y}^{(i)} = \boldsymbol{\Phi}_n\big(\boldsymbol{x}^{(i)}\big), \quad i = 1, \dots, n. \tag{2}$$

We further assume that the map $\boldsymbol{\Phi}_n$ converges toward a limiting continuous function, $\boldsymbol{\Phi} : \mathcal{M} \to \boldsymbol{\Phi}(\mathcal{M})$, when the number of samples goes to infinity. Such limiting maps exist for algorithms such as the Laplacian eigenmaps [2].

In practice, the construction of the map $\boldsymbol{\Phi}_n$ is usually only the first step. Indeed, one is often interested in exploring the configuration space in $\mathbb{R}^d$, and one needs an inverse map to synthesize a new measurement $\boldsymbol{x}$ for a new configuration $\boldsymbol{y} = [\, y_1 \ \cdots \ y_d \,]^T$ in the coordinate domain (see e.g., [7]). In other words, we would like to define an inverse map $\boldsymbol{\Phi}_n^{-1}(\boldsymbol{y})$ at any point $\boldsymbol{y} \in \boldsymbol{\Phi}_n(\mathcal{M})$. Unfortunately, unlike linear methods (such as PCA), nonlinear dimension reduction algorithms only provide an explicit mapping for the original discrete dataset $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$. Therefore, the inverse mapping $\boldsymbol{\Phi}_n^{-1}$ is only defined on these data.

The goal of the present work is to generate a numerical extension of $\boldsymbol{\Phi}_n^{-1}$ to all of $\boldsymbol{\Phi}(\mathcal{M}) \subset \mathbb{R}^d$. To simplify the problem, we assume the mapping $\boldsymbol{\Phi}_n$ coincides with the limiting map $\boldsymbol{\Phi}$ on the data, $\boldsymbol{\Phi}_n(\boldsymbol{x}^{(i)}) = \boldsymbol{\Phi}(\boldsymbol{x}^{(i)})$ for $i = 1, \dots, n$. This assumption allows us to rephrase the problem as follows: we seek an extension of the map $\boldsymbol{\Phi}^{-1}$ everywhere on $\boldsymbol{\Phi}(\mathcal{M})$, given the knowledge that $\boldsymbol{\Phi}^{-1}(\boldsymbol{y}^{(i)}) = \boldsymbol{\Phi}_n^{-1}(\boldsymbol{y}^{(i)}) = \boldsymbol{x}^{(i)}$. We address this problem using interpolation, and we construct an approximate inverse $\boldsymbol{\Phi}_n^{\dagger}$, which converges toward the true inverse as the number of samples, n, goes to infinity,

$$\boldsymbol{\Phi}^{\dagger} : \boldsymbol{\Phi}(\mathcal{M}) \to \mathbb{R}^D, \quad \text{with } \boldsymbol{\Phi}^{\dagger}\big(\boldsymbol{y}^{(i)}\big) = \boldsymbol{x}^{(i)}, \tag{3}$$

$$\text{and} \quad \forall \boldsymbol{y} \in \boldsymbol{\Phi}(\mathcal{M}), \quad \lim_{n\to\infty} \boldsymbol{\Phi}_n^{\dagger}(\boldsymbol{y}) = \boldsymbol{\Phi}^{-1}(\boldsymbol{y}). \tag{4}$$

Using terminology from geometry, we call $\boldsymbol{\Phi}(\mathcal{M})$ the *coordinate domain*, and $\boldsymbol{\Phi}^{-1}$ a *coordinate map* that parametrizes the manifold $\mathcal{M} = \{\boldsymbol{x} \in \mathbb{R}^D; \boldsymbol{x} = \boldsymbol{\Phi}^{-1}(\boldsymbol{y}), \boldsymbol{y} \in \boldsymbol{\Phi}(\mathcal{M})\}$. The components of $\boldsymbol{\Phi}^{-1} = [\, \phi_1^{-1} \ \cdots \ \phi_D^{-1} \,]^T$ are the *coordinate functions*. We note that the focus of the paper is not the construction of new points $\boldsymbol{y}$ in the coordinate domain, but rather the computation of the coordinate functions everywhere in $\boldsymbol{\Phi}(\mathcal{M})$.

2.2. Interpolation of multivariate functions defined on scattered data

Given the knowledge of the inverse at the points $\boldsymbol{y}^{(i)}$, we wish to interpolate $\boldsymbol{\Phi}^{-1}$ over $\boldsymbol{\Phi}(\mathcal{M})$. We propose to interpolate each coordinate function, $\phi_i^{-1}(\boldsymbol{y}), i = 1, \dots, D$ independently of each other. We are thus facing the problem of interpolating a function of several variables defined on the manifold $\boldsymbol{\Phi}(\mathcal{M})$. Most interpolation techniques that are designed for single variable functions can only be extended using tensor products, and have very poor performance in several dimensions. For instance, we know from Mairhuber theorem (e.g., [8]) that we should not use a basis independent of the nodes (for example, polynomial) to interpolate scattered data in dimension $d > 1$. As a result, few options exist for multivariate interpolation. Some of the most successful interpolation methods involve Radial Basis Functions (RBFs) [8]. Therefore, we propose to use RBFs to construct the inverse mapping. Similar methods have been explored in [6,9] to interpolate data on a low-dimensional manifold. We note that while kriging [10] is another common approach for interpolating scattered data, most kriging techniques are equivalent to RBF interpolants [11]. In fact, because in our application we lack specialized information about the covariance structure of the inverse map, kriging is identical to RBF interpolation.

We focus our attention on two basis functions: the Gaussian and the cubic. These functions are representative of the two main classes of radial functions: scale dependent, and scale invariant. In the experimental section we compare the RBF methods to Shepard's method [12], an approach for multivariate interpolation and approximation that is used extensively in computer graphics [13], and which was recently proposed in [14] to compute a similar inverse map.

For each coordinate function ϕ_i^{-1}, we define $\phi_i^{\dagger}$ to be the RBF interpolant to the data $(\boldsymbol{y}^{(j)}, \boldsymbol{x}^{(j)})$,

$$\text{for all } \boldsymbol{y} \in \boldsymbol{\Phi}(\mathcal{M}), \quad \phi_i^{\dagger}(\boldsymbol{y}) = \sum_{j=1}^{n} \alpha_i^{(j)} k\big(\boldsymbol{y}, \boldsymbol{y}^{(j)}\big). \tag{5}$$

The reader will notice that we dropped the dependency on n (number of samples) in $\boldsymbol{\Phi}^{\dagger} = \big[\, \phi_1^{\dagger} \ \cdots \ \phi_D^{\dagger} \,\big]^T$ to ease readability. The function k in (5) is the kernel that defines the radial basis functions, $k(\boldsymbol{z}, \boldsymbol{w}) = g(\|\boldsymbol{z} - \boldsymbol{w}\|)$. The weights, $\{\alpha_i^{(1)}, \dots, \alpha_i^{(n)}\}$,