Missing Data Completion Using Diffusion Maps and Laplacian Pyramids

Neta Rabin and Dalia Fishelov(✉)

Department of Mathematics, Afeka - Tel Aviv Academic College of Engineering, Tel Aviv, Israel
{netar,daliaf}@afeka.ac.il

Abstract. A challenging problem in machine learning is handling missing data, also known as imputation. Simple imputation techniques complete the missing data by the mean or the median values. A more sophisticated approach is to use regression to predict the missing data from the complete input columns. In case the dimension of the input data is high, dimensionality reduction methods may be applied to compactly describe the complete input. Then, a regression from the low-dimensional space to the incomplete data column can be constructed from imputation. In this work, we propose a two-step algorithm for data completion. The first step utilizes a non-linear manifold learning technique, named diffusion maps, for reducing the dimension of the data. This method faithfully embeds complex data while preserving its geometric structure. The second step is the Laplacian pyramids multi-scale method, which is applied for regression. Laplacian pyramids construct kernels of decreasing scales to capture finer modes of the data. Experimental results demonstrate the efficiency of our approach on a publicly available dataset.

Keywords: Missing data · Dimensionality reduction · Diffusion maps · Laplacian pyramids

1 Introduction

A challenging problem in machine learning is preprocessing of the dataset that involves data normalization, detection of input outliers and handling missing data. This work focuses on completion of missing data, also known as imputation. There are several common ways to deal with this problem. Simple imputation techniques complete the missing data by replacing it with the mean or the median value of the column. Another simple imputation approach is to replace the missing values with a sample that is randomly selected from the same column [10,11]. A more sophisticated approach is to use a regression to predict the missing data in specific column from the rest of the columns. In case the dimension of the input data is high, it is reasonable to assume that the data columns are correlated, thus the input matrix $\mathbf{X}$ resides in a low-dimensional space. Therefore, a dimensionality reduction method can be applied to the set

O. Gervasi et al. (Eds.): ICCSA 2017, Part I, LNCS 10404, pp. 284–297, 2017.
DOI: 10.1007/978-3-319-62392-4_21

of full columns and the result may be used as the regression input to predict the missing values in the other columns. Such an approach was suggested in [1] for data imputation in road networks. UshaRani and Sammulal [17] apply dimensionality reduction and clustering for completion of missing medical data. Recently, Pierson and Yau [15] used a linear dimensionality reduction technique to fill in zero-values of single-cell gene expression data.

Dimensionality reduction methods can be separated into feature selection techniques, in which a subset of the columns is selected on the one hand, and feature extraction methods that construct latent combinations of all of the input columns on the other hand. Principal complainant analysis [14] is a common linear dimensionality reduction method that constructs linear combinations of the data columns while minimizing the reconstruction error. However, if the high-dimensional data contains non-linear relations, then linear methods fail to faithfully reduce the dimension of the dataset. Manifold learning methods that include Local Linear Embedding [20], Laplacian Eigenmaps [2,3] and Diffusion Maps [5], aim to reveal the intrinsic parameters that drive the data. These intrinsic modes parameterize the data in a low dimension space while preserving some properties of interest. In this work, diffusion maps are utilized for dimensionally reduction. There, the data is compactly re-organized data according to a diffusion distance metric that is defined by a random walk on the points in the ambient space. In the embedding space, the diffusion distance is the Euclidean distance between the embedded data points, thus the embedding is distance preserving.

Once the dimension of the data is reduced, several methods may be applied for imputations. Linear regression is a simple choice, but it relies on the assumption that there is a linear relationship between the function and the data. Regression using a k nearest neighbors is another simple regression approach, the drawback is that in case different columns need to be imputed, an optimal value for k should be set according to the smoothness of column's data. In this paper, we propose to use Laplcain pyramids for regression from the low-dimensional space to the column with the missing data. The Laplacian pyramids method was proposed in [19] and improved by adding an automatic stopping criteria in [7,9]. The method was applied in [6] for reconstruction data points in the ambient space, in [22] for analog forecasting and in [8] for meteorological data analysis. Here, we emphasize the advantage of this method to automatically stop at a suitable scale that fits the data, without manually tuning the scale parameters. This property is important when running many consecutive regressions for data completion.

2 Mathematical Background

This section describes the two central mathematical tools that are proposed for imputation. First, we review the diffusion maps framework, then the Lalpacian pyramids method is explained.

2.1 Diffusion Maps

Let $\mathbf{X} = \{x_1, \ldots, x_N\}$ be a set of data points in $\mathbb{R}^m$, hence x_i of size $1 \times m$ is the i-th row of $\mathbf{X}$. In order to construct a low-dimensional representation of $\mathbf{X}$, a graph $G = (\mathbf{X}, \mathbf{W})$ is built. The kernel $\mathbf{W}$, defined by $\mathbf{W} = (w(x_i, x_j))_{N\times N}$, contains the weights of the graph edges. We assume that the kernel is

- symmetric;
- positive preserving: $w(x_i, x_j) \geq 0$ for all $x_i, x_j \in \mathbf{X}$;
- positive semi-definite: for all real-valued bounded function f defined on $\mathbf{X}$, $\sum_i \sum_j w(x_i, x_j) f(x_i) f(x_j) \geq 0$.

Diffusion Kernels. Typically, kernels of the form $\mathbf{W}_\epsilon = \left(h\left(\frac{-\|x_i - x_j\|^2}{2\epsilon}\right)\right)$ are chosen since they are directionally independent. Here ϵ defines the width of the kernel. A common choice is the Gaussian kernel $\mathbf{W}_\epsilon = (w_\epsilon(x_i, x_j))$, where $w_\epsilon(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\epsilon}}$. The scale parameter ϵ can be defined (see [21]) by

$$\epsilon = median\{d_{ij}\}, \tag{1}$$

where $\mathbf{D} = (d_{i,j})_{N\times N}$ is the matrix of pairwise Euclidean distances of the set $\mathbf{X}$.

A general normalized form of the kernel, having a parameter α which controls the normalization type, was introduced in [5]. It is given by

$$w_\epsilon^{(\alpha)}(x_i, x_j) = \frac{w_\epsilon(x_i, x_j)}{q^\alpha(x_i) q^\alpha(x_j)}, \qquad q(x_i) = \sum_j w_\epsilon(x_i, x_j). \tag{2}$$

Then, a Markov transition matrix is defined by

$$\mathbf{P}^{(\alpha)} = \left(p^{(\alpha)}(x_i, x_j)\right), \qquad where \qquad p^{(\alpha)}(x_i, x_j) = \frac{w_\epsilon^{(\alpha)}(x_i, x_j)}{\sum_j w_\epsilon^{(\alpha)}(x_i, x_j)} \tag{3}$$

Three values of α that are commonly in use are $\alpha = 0, 1, 0.5$. When $\epsilon \to 0$, $\mathbf{P}^\alpha$ approximates the following operators:

1. $\alpha = 0$: the classical graph Laplacian [4];
2. $\alpha = 1$: the Laplace-Beltrami operator [5];
3. $\alpha = \frac{1}{2}$: the diffussion of the Foller-Planck equation [12].

In this paper α was set to be 1, denoting $\mathbf{P}^{(\alpha=\mathbf{1})}$ as $\mathbf{P}$.

Spectral Decomposition. The construction of the low-dimensional data representation involves the eigendecomposition of $\mathbf{P}$. This is computed by

$$p(x_i, x_j) = \sum_{k\geq 0} \lambda_k \psi_k(x_i) \phi_k(x_j). \tag{4}$$

Here $\{\lambda_k\}_{k=0}^{N-1}$ are the eigenvalues of $\mathbf{P}$ and $\{\phi_k\}_{k=0}^{N-1}$, $\{\psi_k\}_{k=0}^{N-1}$ are the corresponding left and right eigenvectors. We note that $\mathbf{P}$ is similar to a symmetric matrix $\mathbf{A}$, i.e., $\mathbf{A} = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is a diagonal matrix with values $\sum_j w_\epsilon^\alpha(x_i, x_j)$ on its diagonal. Thus, $\mathbf{P}$ and $\mathbf{A}$ share the same set of eigenvalues. The spectral decomposition of the matrix $\mathbf{A} = (a(x_i, x_j))$ is given by

$$a(x_i, x_j) = \sum_{k\geq 0} \lambda_k \mathbf{v}_k(x_i)\mathbf{v}_k(x_j). \tag{5}$$

Since $\mathbf{A}$ is symmetric the set eigenvalues $\{\lambda_k\}_{k=0}^{N-1}$ are real and the set of eigenvectors $\{\mathbf{v}_k\}_{k=0}^{N-1}$ are orthogonal. The left and the right eigenvectors of $\mathbf{P}$ are related to $\{\mathbf{v}_k\}_{k=0}^{N-1}$ by

$$\psi_k = \mathbf{D}^{-\frac{1}{2}}\mathbf{v}_k, \quad \phi_k = \mathbf{D}^{\frac{1}{2}}\mathbf{v}_k. \tag{6}$$

The orthonormality of $\{\mathbf{v}_k\}$ yields the biorthonormality of $\{\phi_k\}$ and $\{\psi_k\}$. This property is used for defining a metric on the data. In addition, since the eigenvalues decay fast to zero, the sum in Eq. (4) can be approximated by a small number of leading terms. These terms are used to define the diffusion maps embedding

$$\Psi(x_i) = (\lambda_1\psi_1(x_i), \lambda_2\psi_2(x_i), \lambda_3\psi_3(x_i), \cdots). \tag{7}$$

Diffusion Distances. This embedding (Eq. (7)) results in a compact representation of the data, in which the distances between the data points are determined by the geometric structure of the data. Following the definitions in [5,13], the diffusion distance between two data points x_i and x_j is the weighted L^2 distance

$$\mathbf{D}^2(x_i, x_j) = \sum_{x_l \in X} \frac{(p(x_i, x_l) - p(x_l, x_j))^2}{\phi_0(x_l)}, \tag{8}$$

where the value of $\frac{1}{\phi_0(x_i)}$ depends on the point's density. In this metric, two data points are close to each other if they are connected by many paths. Substituting Eq. (4) in Eq. (8) and using the biorthogonality properties, we obtain that the diffusion distance is expressed by

$$\mathbf{D}^2(x_i, x_j) = \sum_{k\geq 1} \lambda_k(\psi_k(x_i) - \psi_k(x_j))^2. \tag{9}$$

In these new set of diffusion maps coordinates, the Euclidean distance between two points in the embedded space represents the distances between the points as defined by a random walk.

2.2 The Laplacian Pyramid

The Laplacian pyramid is a multi-scale algorithm for approximating and extending an empirical function f, which is defined on a dataset $\mathbf{Z} = \{z_0, z_1, \ldots, z_n\}$,

to new data points. In this algorithm, Gaussian kernels with descending widths are applied on the points in $\mathbf{Z}$ to construct a multi-resolution approximation of f. Then, this approximation can be extended to evaluate f for new points $\{\bar{z}\}$. An initial Gaussian kernel, having a relatively large scale σ_0, is defined on $\mathbf{Z}$ by $\mathbf{G_0} = (g_0(z_i, z_j))$ where

$$g_0(z_i, z_j) = e^{\frac{-\|z_i - z_j\|^2}{\sigma_0}}, \qquad z_i, z_j \in \mathbf{Z}. \tag{10}$$

Normalizing $\mathbf{G}_0$ results in a smoothing operator $\mathbf{K}_0 = (k_0(z_i, z_j))$, where

$$k_0(z_i, z_j) = q_0^{-1}(z_i) g_0(z_i, z_j), \quad \text{where} \quad q_0(z_i) = \sum_j g_0(z_i, z_j). \tag{11}$$

At a finer scale l, the Gaussian kernel $\mathbf{G}_l = (g_l(z_i, z_j))$ is defined by

$$g_l(z_i, z_j) = e^{-\|(z_i - z_j)\|^2 / (\frac{\sigma_0}{2^l})}, \qquad l = 1, 2, 3, \ldots \tag{12}$$

Normalization of $\mathbf{G}_l$ yields the smoothing operator $\mathbf{K}_l = (k_l(z_i, z_j))$, where

$$k_l(z_i, z_j) = q_l^{-1}(z_i) g_l(z_i, z_j), \qquad q_l(z_i) = \sum_j g_l(z_i, z_j), \qquad l = 1, 2, 3, \ldots \tag{13}$$

The Laplacian Pyramid representation of f is iteratively defined as follows. For the first level $l = 0$, a smooth approximation of f is

$$f_0(z_k) = \sum_{i=1}^{n} k_0(z_k, z_i) f(z_i), \qquad k = 1, \ldots, n, \qquad z_i, z_k \in Z. \tag{14}$$

Let

$$d_1(z_i) = f(z_i) - f_0(z_i), \qquad i = 1, 2, \ldots, n \qquad z_i \in \mathbf{Z},$$

then a finer representation of f is

$$f_1(z_k) = f_0(z_k) + \sum_{i=1}^{n} k_1(z_k, z_i) d_1(z_i), \qquad k = 1, \ldots, n.$$

In general, for $l = 1, 2, 3 \ldots$,

$$d_l(z_i) = f(z_i) - f_{l-1}(z_i), \qquad i = 1, \ldots, n, \tag{15}$$

$$f_l(z_k) = f_{l-1}(z_k) + \sum_{i=1}^{n} k_l(z_k, z_i) d_l(z_i), \qquad k = 1, \ldots, n, \tag{16}$$

where f_0 is defined in Eq. (14). Equation (16) approximates a given function f by the series of functions $\{f_0, f_1, f_2, \ldots\}$ in a multi-scale manner, going from a coarser to a finer representation. The functions $\{f_0, f_1, f_2, \ldots\}$ can be easily extended to a new point $\bar{z}$ in the following way.

$$f_0(\bar{z}) = \sum_{i=1}^{n} k_0(\bar{z}, z_i) f(z_i) \qquad \text{for} \quad l = 0 \tag{17}$$

$$f_l(\bar{z}) = f_{l-1}(\bar{z}) + \sum_{i=1}^{n} k_l(\bar{z}, z_i) d_l(z_i) \qquad \text{for } l = 1, 2, 3, \ldots, \tag{18}$$

where $d_l(z_i)$ is defined in Eq. (15).

The following example (taken from [6]) demonstrates the multi-scale approximation of the function

$$f(x) = \begin{cases} -0.02(x-4\pi)^2 + sin(x) & 0 \le x \le 4\pi \\ -0.02(x-4\pi)^2 + sin(x) + \frac{1}{2}sin(3x) & 4\pi < x \le 7.5\pi \\ -0.02(x-4\pi)^2 + sin(x) + \frac{1}{2}sin(3x) + \frac{1}{4}sin(9x) & 7.5\pi < x \le 10\pi \end{cases} \tag{19}$$

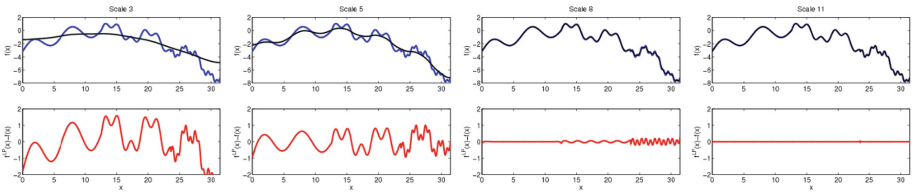

Fig. 1. Approximations of the function f that was defined in Eq. (19) for scales $l = 3, 5, 8, 11$ going from left to right. The function is plotted in blue in each of the top images, approximations f_l in black and the corresponding residuals d_l on the bottom row in red. (Color figure online)

Stopping Criteria and the Auto-adaptive Laplacian Pyramids. The Laplacian Pyramids iterations may be stopped by setting an admissible error to a small threshold *err*, for example by requiring $\|f - f_l\| < err$. When *err* is too large, then the iterations stop at a coarse scale, thus the approximation does not capture finer structures of the function f. If *err* is too small, then in finer scales a point may have few or no neighbors, thus over-fitting may occur. The auto-adaptive Laplacian Pyramids, which were introduced in [7,9], slightly modify the kernels constructed in Eqs. (10) and (12). This prevents over-fitting and provides a criteria for selecting a proper stopping scale l. The main modification is to replace the kernels $\mathbf{G}_l = (g_l(z_i, z_j))$ by $\tilde{\mathbf{G}}_l$, which are defined by

$$\tilde{\mathbf{G}}_l(z_i, z_j) = \begin{cases} \mathbf{G}_l(z_i, z_j) & i \neq j \\ 0 & i = j. \end{cases} \tag{20}$$

These yield the normalized operators $\tilde{k}_l(z_i, z_j) = \tilde{q}_l^{-1}(z_i)\tilde{g}_l(z_i, z_j)$, where $\tilde{q}_l(z_i) = \sum_j \tilde{g}_l(z_i, z_j)$ and the iterative construction

$$f_0(z_k) = \sum_{i=1}^{n} \tilde{k}_0(z_k, z_i) f(z_i) \qquad \text{for level} \quad l = 0 \tag{21}$$

$$f_l(z_k) = f_{l-1}(z_k) + \sum_{i=1}^{n} \tilde{k}_l(z_k, z_i) d_l(z_i) \qquad \text{for} \qquad l = 1, 2, \ldots. \tag{22}$$

Extension to new points is done in a similar manner, $\bar{z}$ replaces z_k in Eqs. (21) and (22).

By using the above modification, the pyramids are constructed using a Leave-one-out-cross-validation that is inherent in the algorithm, as each train point in $\mathbf{Z}$ is treated as test point. The approximation of f at z_i is built without using the value of the point itself, the contribution is only from $z_i's$ neighboring points. This modification makes the procedure more robust in the presence of noise. The stopping scale l is determined by computing the mean square error $err_l = \|f - f_l\|$ at each level and choosing the stopping scale l as the minimum value of the vector err_l. To conclude, this procedure is equivalent to running the Laplacian Pyramids algorithm in a Leave one out cross validation manner and choosing the scale where the error is minimal.

3 Imputation via Diffusion Maps and Adaptive Laplacian Pyramids

This section describes how diffusion maps and Laplacian pyramids are used for completing values of missing data. Let $\mathbf{X}$ be the input data matrix of size $N \times m$. For example, in a medical application N represents the number of patients and m is the number of medical results for each patient. For simplicity, we assume that missing data occurs in a single column $\mathbf{X}^{(j)}$ of the input matrix $\mathbf{X}$, and that the other columns $\{\mathbf{X}^{(k)}\}_{k \neq j}$ are complete. Thus, one can regress $\mathbf{X}^{(j)}$ using $\{\mathbf{X}^{(k)}\}_{k \neq j}$ or its corresponding low-dimensional representation.

Figure 2 provides an illustrative description of our approach. The input matrix $\{\mathbf{X}^{(k)}\}_{k \neq j}$ is given on the left of the figure in black. It is of dimension $N \times (m-1)$. The blue column next to it, $\mathbf{X}^{(j)}$, is of dimension $N \times 1$. Its red circles represent missing data while the blue line markers represent complete values. In accordance, the black lines in the matrix $\{\mathbf{X}^{(k)}\}_{k \neq j}$ indicate rows that have a value in $\mathbf{X}^{(j)}$, while the red lines correspond to rows with missing data in $\mathbf{X}^{(j)}$. In the center of the figure, the diffusion maps representation of the matrix $\{\mathbf{X}^{(k)}\}_{k \neq j}$ is plotted. Here, the black and red circles represent the embedding of black and red lines of the matrix $\{\mathbf{X}^{(k)}\}_{k \neq j}$, accordingly. On the right side of the figure, the column $\mathbf{X}^{(j)}$ is plotted as a function that is defined on the embedded points. Laplacian pyramids are constructed using the low-dimensional model on the right.

The proposed algorithm consist of the following three steps, a pre-processing step (Step 0) is also described.

Step 0: Preprocessing.

- For the simplicity of notations, sort the N points in $\mathbf{X}$ so that the first n, $n < N$ points are those that have a value in $\mathbf{X}^{(j)}$ and the following $N - n$ points have missing values in $\mathbf{X}^{(j)}$.
- In order to apply diffusion maps, the column of the input data matrix $\{\mathbf{X}^{(k)}\}_{k \neq j}$ should be of a comparable scale. Otherwise, if columns with large values dominate the pairwise distanced that are constructed in the kernel (see Sect. 2.1). A simple scale normalization can be performed by subtracting the

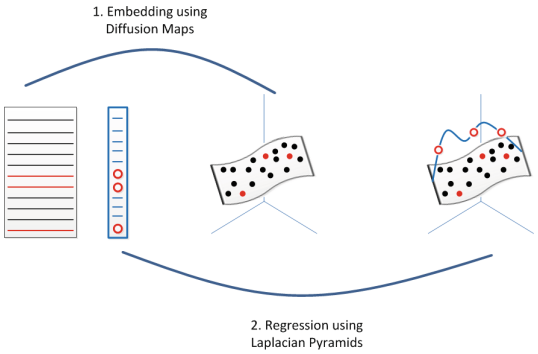

Fig. 2. Illustrative description of the proposed algorithm.

mean and dividing by the variance of each column. Another simple approach is to divide each column by it norm. A more sophisticated approach, which is based on diffusion kernels, was proposed in [18,19]. For the next step, we assume $\{\mathbf{X}^{(k)}\}_{k\neq j}$ holds columns of comparable scales.

Step 1: Reduce the dimension of the set $\{\mathbf{X}^{(k)}\}_{k\neq j}$ containing N rows and m columns via diffusion maps.

1. Construct a kernel

$$\mathbf{W}_\epsilon = w_\epsilon(x_i, x_j)) = e^{\frac{-\|x_i - x_j\|^2}{2\epsilon}}.$$

The scale parameter ϵ can be computed as explained in Sect. 2.1. This results in an $N \times N$ symmetric matrix.

2. Use the Laplace-Bletrami normalization (see Sect. 2.1) and build

$$\mathbf{W}_\epsilon^{(1)} = w_\epsilon^{(1)}(x_i, x_j) = \frac{w_\epsilon(x_i, x_j)}{q(x_i)q(x_j)}, \qquad q(x_i) = \sum_j w_\epsilon(x_i, x_j).$$

3. Apply a row-normalization and obtain the Markov matrix

$$\mathbf{P} = p(x_i, x_j) = \frac{w_\epsilon^1(x_i, x_j)}{\sum_j w_\epsilon^1(x_i, x_j)}.$$

4. Compute the spectral decomposition of $\mathbf{P}$,

$$\mathbf{P} = p(x_i, x_j) = \sum_{k\geq 0} \lambda_k \psi_k(x_i)\phi_k(x_j).$$

5. Use the first $d << N$ leading diffusion coordinates

$$\Psi(x_i) = (\lambda_1\psi_1(x_i), \ldots, \lambda_d\psi_d(x_i)).$$

to embed $\{\mathbf{X}^{(k)}\}_{k\neq j}$ in a d-dimensional space.

Step 2: Set $f = \mathbf{X}^{(j)}$ and approximate from the corresponding points in the low-dimensional space.

1. Collect the subset of points from the embedding space that have a value in f. Denote these n point ($n < N$) by $\mathbf{Z} = \{\Psi(x_i)\}_{i=1}^{n}$, where x_i are points for which $f(x_i) = \mathbf{X}^{(j)}(i)$ is known.
2. Collect the subset of points from the embedding space that have a missing value in f. Denote these $N-n$ points $\bar{\mathbf{X}} = \{\Psi(x_i)\}_{i=N-n+1}^{N}$, x_i are points for which $\mathbf{X}^{(j)}(i)$ needs to be imputed.
3. Compute the auto-adaptive Laplacian pyramids for the function $f = \mathbf{X}^{(j)}$ using the points in Z by

$$f_0(z_k) = \sum_{i=1}^{n} \tilde{k}_0(z_k, z_i) f(z_i) \qquad \text{for} \quad l = 0$$

$$f_l(z_k) = f_{l-1}(z_k) + \sum_{i=1}^{n} \tilde{k}_l(z_k, z_i) d_l(z_i) \qquad \text{for} \quad l = 1, 2, \ldots$$

4. In each level store the error $err_l = \|f - f_l\|$.
5. Set the final (stopping) level $\tilde{l}$ to be $\tilde{l} = min(err_l)$.

Step 3: For each point $\bar{z} \in \bar{\mathbf{Z}}$, impute its value $f(\bar{z})$, where $f = \mathbf{X}^{(j)}$ by

$$f_0(\bar{z}) = \sum_{i=1}^{n} \tilde{k}_0(\bar{z}, z_i) f(z_i) \qquad \text{for} \quad l = 0$$

$$f_l(\bar{z}) = f_{l-1}(\bar{z}) + \sum_{i=1}^{n} \tilde{k}_l(\bar{z}, z_i) d_l(z_i) \qquad \text{for} \quad l = 1, 2, \ldots, \tilde{l}.$$

Set the missing values $f(\bar{z})$ to be $f_l(\bar{z})$.

4 Experimental Results

The application of the proposed algorithm is demonstrated on a mice protein expression dataset, taken from the UCI repository [16]. The data set consists of the expression levels of 77 proteins that produced detectable signals in the nuclear fraction of cortex. These types of biological datasets, including the above dataset, often have missing values. From this dataset, we select a subset of $N = 1000$ rows and $m = 66$ columns, which do not contain missing data. Assuming now that some data is missing in a particular column, then diffusion maps followed by Laplacian pyramids (see Sect. 3) is applied for imputation. We measure the efficiently of the proposed algorithm by calculating the mean and maximum errors. We compare our results to linear regression and K-nearest neighbors techniques. The evaluation is carried out using a 5-fold cross validation.

Let $\{\mathbf{X}^{(k)}\}_{k \neq 10}$ be the full dataset and $\mathbf{X}^{(10)}$ be a column with missing data. First, the data set $\{\mathbf{X}^{(k)}\}_{k \neq 10}$ is pre-processed according to Step 0 in Sect. 3 by substraction of the mean and division by the standard deviation of each

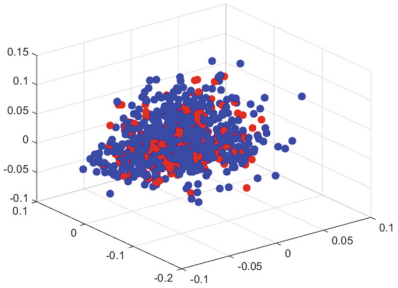

Fig. 3. Diffusion maps embedding of the set $\{\mathbf{X}^{(k)}\}_{k\neq 10}$. Points with known values in $\mathbf{X}^{(10)}$ are colored blue and points with missing values in $\mathbf{X}^{(10)}$ are colored red. (Color figure online)

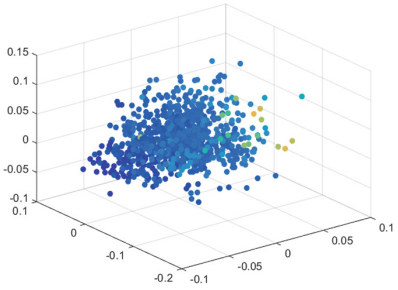

Fig. 4. Diffusion maps embedding of the set $\{\mathbf{X}^{(k)}\}_{k\neq 10}$ colored by the values of $f = \mathbf{X}^{(10)}$.

column. Next, diffusion maps is applied to the set $\{\mathbf{X}^{(k)}\}_{k\neq 10}$ and embedded in a 3-dimensional space. The values of $\mathbf{X}^{(10)}$ are known for $n = 800$ points and unknown for the remaining $N - n = 200$ points. Figure 3 presents the 3-dimensional embedding of $\{\mathbf{X}^{(k)}\}_{k\neq 10}$. The 800 points for which the values of $\mathbf{X}^{(10)}$ are known are colored in blue where as the 200 points that have missing values in $\mathbf{X}^{(10)}$ are colored in red. Coloring the embedding by the values of the function $f = \mathbf{X}^{(10)}$, we can induce from Fig. 4 that the function is smooth on the embedded data.

Next, the adaptive Laplacian pyramids are constructed with $n = 800$ known values. The stopping level $\tilde{l}$ for the Laplacian pyramids is set to be the iteration having the minimum error (as explained in Sect. 2.2). Figure 5 displays the values of err_l as computed for each iteration of the Laplacian pyramids; the minimum is reached for $l = 14$. This is the stopping scale used for extending the pyramids (see Step 3 in Sect. 3). Last, the $N - n = 200$ missing values are imputed with Laplacian pyramids and compared to linear regression and k-nearest neighbors. All three methods rely on the embedded data instead of the original high-dimensional space. Figure 6 displays the approximated values as imputed by the Laplacian pyramids (in blue), linear regression (in green) and k-nn with $k = 7$ (in yellow). The error bars as computed from a 5-fold cross validation are presented in Fig. 7. We plot the mean square error in the left

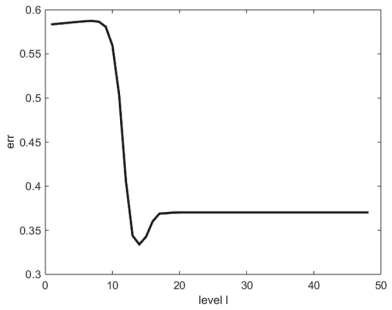

Fig. 5. Approximation errors for each scale l in the Laplacian pyramid construction. The stopping scale is set to be the level $\tilde{l}$ with the minimal error, here $\tilde{l} = 14$.

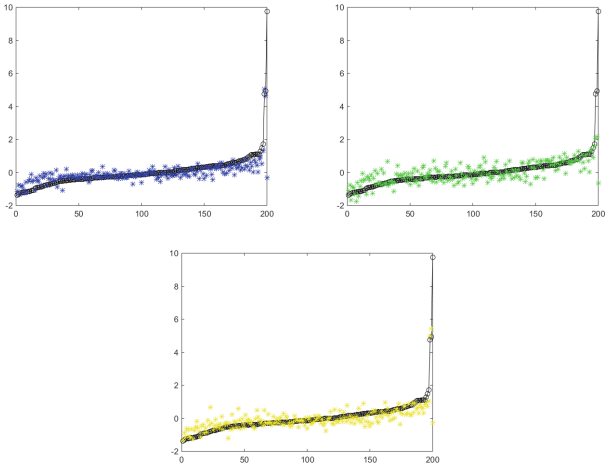

Fig. 6. Imputation of $f = \mathbf{X}^{(10)}$ (the missing values are sorted) using the adaptive Laplacian pyramids (blue), linear regression (green) and k-nn, $k = 7$ (yellow). The true values are in black. (Color figure online)

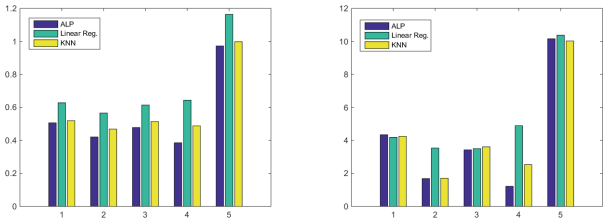

Fig. 7. Mean squared error (left) and maximum error (right) for the 5-fold cross validation imputation of $f = \mathbf{X}^{(10)}$.

chart and the maximum error in the right chart. It can be seen that the linear regression results with the largest error and that the pyramids based imputation yields the smallest errors.

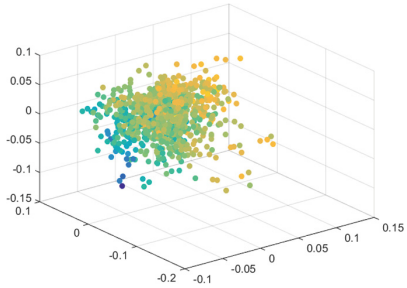

Fig. 8. Diffusion maps embedding of the set $\{\mathbf{X}^{(k)}\}_{k\neq 48}$ colored by the values of $f = \mathbf{X}^{(48)}$.

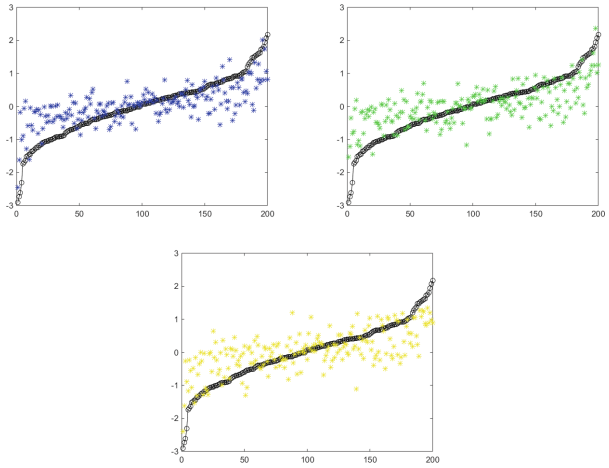

Fig. 9. Imputation of $f = \mathbf{X}^{(48)}$ (the missing values are sorted) using the adaptive Laplacian pyramids (blue), linear regression (green) and k-nn, $k = 7$ (yellow). The true values are in black. (Color figure online)

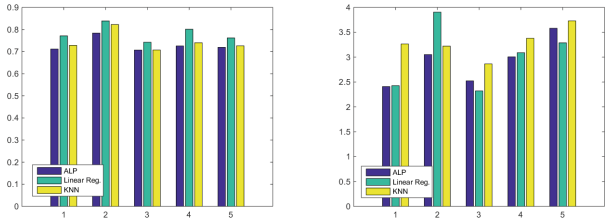

Fig. 10. Mean squared error (left) and maximum error (right) for the 5-fold cross validation imputation of $f = \mathbf{X}^{(48)}$.

We repeat the above example for another column $\mathbf{X}^{(48)}$. As before, the set $\{\mathbf{X}^{(k)}\}_{k\neq 48}$ is embedded into a 3-dimensional space. The function $f = \mathbf{X}^{(48)}$ is not as smooth as $\mathbf{X}^{(10)}$, this is displayed in Fig. 8, where the diffusion maps

embedding is colored by $f = \mathbf{X}^{(48)}$. Figure 9 plots the imputation results for the missing values (sorted) as evaluated by the adaptive Laplacian pyramids, linear regression and k-nn with $k = 7$. The means square errors and the maximum errors are presented in Fig. 10. The pyramids approximations maintain small mean and max errors throughout all of the 5 folds.

5 Conclusions

In this paper, a two-step method was presented for data completion, which is suitable for high dimensional data. The proposed algorithm uses diffusion maps for reducing the dimension of the training samples with complete data in the first step. Next, a regression process is carried out in order to evaluate missing values from a particular column. The regression analysis is done with a multi-scale method named adaptive Laplacian pyramids that learns the suitable scale for regression. The method is not sensitive to the choice of parameters. Since in such imputation problems, regression is applied many times for filling in data from different columns, such properties are important. Experimental results show the advantages of the proposed method compared to linear regression and local k-nearest neighbors regression.

References

1. Asif, M.T., Mitrovic, N., Garg, L., Dauwels, J., Jaillet, P.: Low-dimensional models for missing data imputation in road networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3527–3531 (2013)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**, 1373–1396 (2003)
3. Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds. Mach. Learn. **56**, 209–239 (2004)
4. Chung, F.R.K.: Spectral Graph Theory. AMS Regional Conference Series in Mathematics (1997)
5. Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmon. Anal. **21**, 5–30 (2006)
6. Dsilva, C.J., Talmon, R., Rabin, N., Coifman, R.R., Kevrekidis, I.G.: Nonlinear intrinsic variables and state reconstruction in multiscale simulations. J. Chem. Phys. **139**(18), 184109 (2013)
7. Fernández, Á., Rabin, N., Fishelov, D., Dorronsoro, J.R.: Auto-adaptative laplacian pyramids for high-dimensional data analysis. arXiv preprint arXiv:1311.6594
8. Fernández, Á., González, A.M., Díaz, J., Dorronsoro, J.R.: Diffusion maps for dimensionality reduction and visualization of meteorological data. Neurocomputing **163**, 25–37 (2015)
9. Fernández, Á., Rabin, N., Fishelov, D., Dorronsoro, J.R.: Auto-adaptive Laplacian Pyramids. In: 24th European Symposium on Artificial Neural Networks. Computational Intelligence and Machine Learning, ESANN, pp. 59–64, Bruges, Belgium (2016)
10. Huisman, M.: Missing data in behavioral science research: investigation of a collection of data sets. Kwant. Methoden **57**, 69–93 (1998)

11. Little, J.A.R., Rubin, B.D.: Statistical Analysis with Missing Data, 2nd edn. Wiley, Hoboken (2002)
12. Nadler, B., Lafon, S., Coifman, R.R., Kevrekidis, I.G.: Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In: Neural Information Processing Systems (NIPS), vol. 18 (2005)
13. Nadler, B., Lafon, S., Coifman, R.R., Kevrekidis, I.G.: Diffusion maps, spectral clustering and reaction coordinate of dynamical systems. Appl. Comput. Harmon. Anal. **21**, 113–127 (2006)
14. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**(11), 559–572 (1901)
15. Pierson, E., Yau, C.: ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. **16**, 241 (2015)
16. http://archive.ics.uci.edu/ml/datasets
17. UshaRani, Y., Sammulal, P.: An efficient disease prediction and classification using feature reduction based imputation technique. In: International Conference on Engineering & MIS (ICEMIS) (2016)
18. Rabin, N., Averbuch, A.: Detection of anomaly trends in dynamically evolving systems. In: 2010 AAAI Fall Symposium Series, pp. 44–49 (2010)
19. Rabin, N., Coifman, R.R.: Heterogeneous datasets representation and learning using diffusion maps and Laplacian pyramids. In: Proceedings of the 2012 SIAM International Conference on Data Mining, pp. 189–199 (2012)
20. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
21. Schclar, A.: A diffusion framework for dimensionality reduction. In: Maimon, O., Rokach, L. (eds.) Soft Computing for Knowledge Discovery and Data Mining, pp. 315–325. Springer, Heidelberg (2008). doi:10.1007/978-0-387-69935-6_13
22. Zhao, Z., Giannakis, D.: Analog forecasting with dynamics-adapted kernels. Nonlinearity **29**, 2888 (2016)