A multi-scale approach for data imputation

Neta Rabin
Department of Mathematics
Afeka - Tel-Aviv Academic College of Engineering, Israel
Tel-Aviv, 6910717
netar@afeka.ac.il

Dalia Fishelov
Department of Mathematics
Afeka - Tel-Aviv Academic College of Engineering, Israel
Tel-Aviv, 6910717
daliaf@afeka.ac.il

***Abstract*—A common pre-possessing task in machine learning is to complete missing data entries in order to form a full dataset. In case the dimension of the input data is high, it is often the case that the rows and columns are correlated. In this work, we construct a multi-scale model that is based on the the dual row-column geometry of the dataset and apply it to imputation. The imputation is carried out within the model construction. Experimental results demonstrate the efficiency of our approach on a publicly available dataset.**

I. Introduction

Completion of missing data, which is also known as imputation, is a common pre-processing task that arises in signal processing, data mining and machine learning applications. Simple imputations approaches propose to impute the missing data with the mean or median value that is calculated from the known data entities. When the data has a rectangular structure of row and columns, the mean and median based imputations may rely on the known values in the same row or column of the missing value [6], [8]. More sophisticated methods for imputation use random values that are drawn from a distribution that fits to the known data values. For example the Multiple imputation algorithm [12] uses a posterior distribution of the data to draw multiple values for each missing entry and the final value is determined by pooling step that chooses one value based on the mean and variance. Another approach is to use regression to complete missing data. For a given column with missing values, the column is regressed against other columns for which the values are known and the regression based predictions impute the missing data.

When the dimension of the input data is high, and in particular when each data instance (data row) consist of a large number of parameters, it is often the case that the columns reside in a low-dimensional space. The low dimensional representation of the column space can be used to construct regression type models for imputing missing values. In [1], the authors suggested an approach that constructs a low-dimensional model for data imputation in road networks. Dimensionality reduction and clustering was applied for imputation of medical data [14]. Recently, Pierson and Yau [9] used a linear dimensionality reduction technique to fill in zero-values of single-cell gene expression data and [15] proposed a diffusion maps based imputation method for gene-gene iterations.

In previous work [11], we proposed a two-step algorithm for data regression based imputation. The first step utilizes a non-linear manifold learning technique named diffusion maps [2] for reducing the dimension of the data in terms of columns. Diffusion maps faithfully embeds complex data while preserving its geometric structure. The second regression step was based on the Laplacian pyramids multi-scale method [10]. Laplacian pyramids construct kernels of decreasing scales to capture finer modes of the data and the scale is automatically fit to the data density and noise. When regression is carried out for imputation in a large number of columns, the main advantage of this scheme is the automatic scale adaption that is fitted to the behavior of the data in each column [4], [5]. However, when the number of rows and the number of columns are both large, column-based regression methods for imputation do not fully take into account the connections in the data, they mainly relay on the connections between the rows.

In this work, the dual geometry structure of the dataset is utilized by modeling the rows and the columns alternately in a multi-scale manner. The multi-scale construction is achieved by extending Laplacian pyramids technique to work in a two-directional mode, this is described in Section II. Laplacian pyramids are reviewed in Section II-A. The two-directional Laplacian pyramids algorithm is described in Section II-B. In Section III we explains how the proposed method is applied for imputation. Finally, experimental results are provided in Section IV, these demonstrate the efficiency of our approach on a synthetic example and a publicly available dataset.

II. Methods

A. One-directional Laplacian Pyramids

The Laplacian pyramids is a multi-scale algorithm for approximating and extending an empirical function f, which is defined on a dataset $Z = \{z_0, z_1, \ldots, z_n\}$, to new data points. In this algorithm, Gaussian kernels with descending widths are applied on the points in 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, $\mathbf{G_0}$, having a relatively large scale σ_0, is defined on Z by

$$g_0(z_i, z_j) = e^{\frac{-\|z_i - z_j\|^2}{\sigma_0}}, \qquad z_i, z_j \in Z. \tag{1}$$

978-1-5386-6378-3/18/$31.00

Normalizing $\mathbf{G_0}$ results in a smoothing operator

$$\mathbf{K_0} = (k_0(z_i, z_j)) = q_0^{-1}(z_i) g_0(z_i, z_j), \tag{2}$$

where $q_0(z_i) = \sum_j g_0(z_i, z_j)$. At a finer scale l, the Gaussian kernel $\mathbf{G_l}$ is defined by

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

Normalization of $\mathbf{G_l}$ yields the smoothing operator

$$\mathbf{K_l} = k_l(z_i, z_j) = q_l^{-1}(z_i) g_l(z_i, z_j), \tag{3}$$

where $\qquad q_l(z_i) = \sum_j g_l(z_i, z_j), \qquad l = 1, 2, 3, \ldots$

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

$$f_0(z_k) = \sum_{i=1}^{n} k_0(z_k, z_i) f(z_i), \quad k = 1, \ldots, n, \quad z_i, z_k \in Z. \tag{4}$$

Let

$$d_1(z_i) = f(z_i) - f_0(z_i), \quad i = 1, 2, \ldots, n \quad z_i \in Z,$$

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 $l = 1, 2, 3 \ldots$,

$$d_l(z_i) = f(z_i) - f_{l-1}(z_i), \qquad i = 1, \ldots, n, \tag{5}$$

$$f_l(z_k) = f_{l-1}(z_k) + \sum_{i=1}^{n} k_l(z_k, z_i) d_l(z_i), \qquad k = 1, \ldots, n, \tag{6}$$

where f_0 is defined in Equation (4). Equation (6) 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.

$$f_0(\bar{z}) = \sum_{i=1}^{n} k_0(\bar{z}, z_i) f(z_i) \qquad \text{for} \quad l = 0 \tag{7}$$

$$f_l(\bar{z}) = f_{l-1}(\bar{z}) + \sum_{i=1}^{n} k_l(\bar{z}, z_i) d_l(z_i) \qquad \text{for } l = 1, 2, 3, \ldots, \tag{8}$$

where $d_l(z_i)$ is defined in Equation (5).

The following example (taken from [3]) demonstrates the multi-scale approximation of the function that has in it three different frequencies that are added to $h(x) = -0.02(x - 4\pi)^2$.

$$f(x) = \begin{cases} h(x) + sin(x), x \in I_1 \\ h(x) + sin(x) + \frac{1}{2} sin(3x), x \in I_2 \\ h(x) + sin(x) + \frac{1}{2} sin(3x) + \frac{1}{4} sin(9x), x \in I_3 \end{cases} \tag{9}$$

The regions I_1, I_2 and I_3 are defined by

$$\begin{aligned} I_1 &= 0 \le x \le 4\pi \\ I_2 &= 4\pi < x \le 7.5\pi \\ I_3 &= 7.5\pi < x \le 10\pi. \end{aligned} \quad . \tag{10}$$

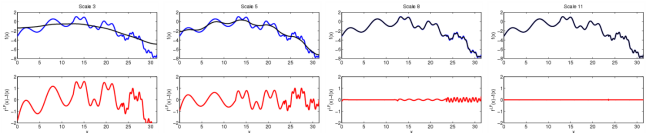

Fig. 1. Approximations of the function f that was defined in Equation (9) for scales $l = 3, 5, 8, 11$ going from left to right. The function is plotted in blue in each of the top images, approximations f_l in black and the corresponding residuals d_l on the bottom row in red.

Figure 1 demonstrates how the function is approximated in a multi-scale manner.

1) 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 [5], [4], slightly modify the kernels constructed in Equations (2) and (3). This prevents over-fitting and provides a criteria for selecting a proper stopping scale l. The main modification is to replace the kernels $G_l = (g_l(z_i, z_j))$ by $\tilde{G}_l$, which are defined by

$$\tilde{\mathbf{G}}_\mathbf{l}(z_i, z_j) = \begin{cases} \mathbf{G_l}(z_i, z_j) & i \neq j \\ 0 & i = j. \end{cases} \tag{11}$$

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

$$f_0(z_k) = \sum_{i=1}^{n} \tilde{k}_0(z_k, z_i) f(z_i) \qquad \text{for level} \quad l = 0 \tag{12}$$

$$f_l(z_k) = f_{l-1}(z_k) + \sum_{i=1}^{n} \tilde{k}_l(z_k, z_i) d_l(z_i) \qquad \text{for} l = 1, 2, \ldots. \tag{13}$$

Extension to new points is done in a similar manner, $\bar{z}$ replaces z_k in Equations (12) and (13).

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 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.

B. Two-directional Laplacian Pyramids

Given a function $f = f(x, y)$ of size $M \times N$, the Laplacian pyramid method is modified to work in a two-direction manner that takes into account the relationship between the rows and columns of the data. At each scale l, two kernels are constructed. The kernels are based on the pairwise distances between the rows and columns of f, and are denoted by $\mathbf{G}_l^{(L)}$ and $\mathbf{G}_l^{(R)}$, respectively, where L stands for left and R stands for right. Denote the associated normalized kernels by $\mathbf{K}_l^{(L)}$ and $\mathbf{K}_l^{(R)}$. First, f is coarsely approximated by

$$f_0 = \mathbf{K}_0^{(L)} * f * \mathbf{K}_0^{(R)}.$$

Next, the difference

$$d_1 = f - f_0,$$

is calculated and it becomes input for the next finer approximation. In the second step f is approximated by

$$f_1 = f_0 + \mathbf{K}_1^{(L)} * d_1 * \mathbf{K}_1^{(R)}.$$

After l iterations the difference between f and its multi-scale representation is given by

$$d_l = f - f_{l-1},$$

and the a finer version of f is

$$f_l = f_{l-1} + \mathbf{K}_l^{(L)} * d_l * \mathbf{K}_l^{(R)}. \tag{14}$$

III. Application to Data Imputation

Let X be a dataset of size $M \times N$ that has in it missing values. The two-directional Laplacian pyramids is applied for approximating X is a multi-scale manner. Begin by constructing a normalized row-kernel, denoted by $\mathbf{K}_0^{(L)}$, based on the known data entries in the rows X. For example, if $\mathbf{K}_0^{(L)}$ is a Gaussian kernel, then the distances between two rows $X(i,:)$ and $X(j,:)$ that belong to X can be computed by considering only the subset of columns in which the two rows both have known values. The distance $e^{\frac{-\|X(i,:)-X(j,:)\|^2}{\sigma_0}}$ is then evaluated restricted to this set of columns. Next, X is convolved with the wide row kernel and the result $\mathbf{K}_0^{(L)} * X$ is a smooth version of X that has values for all entries of X, in particular where data was missing. Similarly, a course column-kernel is constructed based on the pariwise distances between the columns of X. The pairwise distances $e^{-\frac{\|X(:,i)-X(:,j)\|^2}{\epsilon_0}}$ are calculated restricted to the known data entries. A first course and imputed approximation of X is then

$$X_0 = \mathbf{K}_0^{(L)} * X * \mathbf{K}_0^{(R)}.$$

The residual $D_1 = X - X_0$ is input for the next iterations. The kernel scales are modified to be $\sigma_1 = \frac{\sigma_0}{2}$ and $\epsilon_1 = \frac{\epsilon_0}{2}$ and new kernels $\mathbf{K}_1^{(L)}$ and $\mathbf{K}_1^{(R)}$ are computed. By convolving D_1 with the new kernels, a finer, imputed representation of X,

$$X_1 = X_0 + \mathbf{K}_1^{(L)} * D_1 * \mathbf{K}_1^{(R)}$$

is obtained.

The iterations continue for a pre-defined number of times, and the optimal stopping scale $L = L^\star$ is set automatically as explained in Section II-A1. At each step we have

$$X_L = X_{L-1} + \mathbf{K}_L^{(L)} * D_{L-1} * \mathbf{K}_L^{(R)}.$$

The process results in the imputed multi-scale representation X that is given by

$$X_{L^\star} = X_0 + \sum_{l=1}^{L^\star} \mathbf{K}_l^{(L)} * D_{l-1} * \mathbf{K}_l^{(R)}. \tag{15}$$

IV. Experimental Results

In this section we provide two examples that demonstrate the described approach. The first, synthetic example, holds sample values from the function $f(x, y) = sin(4x)sin(y)$, $0 \le x \le 2\pi$, $0 \le y \le \pi$. The data is of size $M \times N = 120 \times 60$ and has in it 20% missing values. Note that the dataset X is just the function values, $X = f(x, y)$. Figure 2 displays the function with missing values.

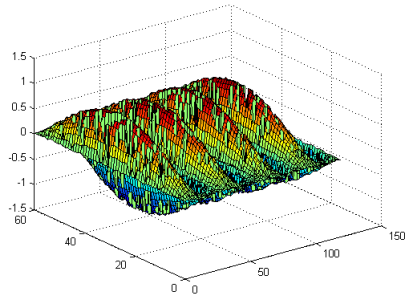

Fig. 2. $f(x, y) = sin(4x)sin(y)$ with 20% of missing values.

The multi-scale imputation process stops after 5 iterations. The imputed approximations X_0, X_1, X_3, and X_5, are plotted in Figure 3. The root mean square error for the imputed data is 0.0203.

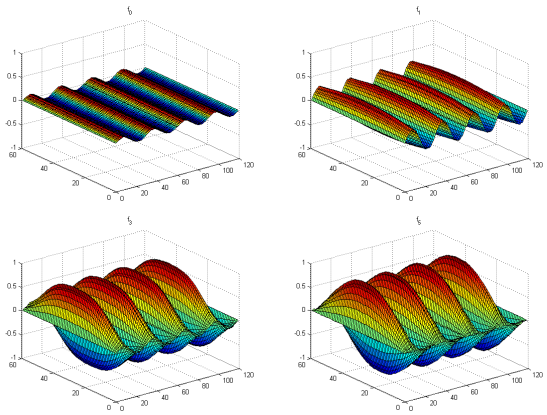

Fig. 3. Top left: X_0, top right: X_1, bottom left: X_3 and bottom right: X_5 - the finest approximation level.

The same procedure is carried out once more with 80% missing data. The function is plotted in Figure 4. The multi-scale process stops after 7 iterations, imputed approximations X_1, X_4, X_6, and X_7, are plotted in Figure 5. The root mean square error for the imputed data is 0.1169.

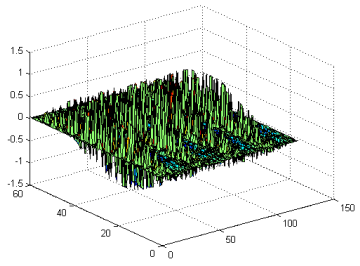

Fig. 4. $f(x, y) = sin(4x)sin(y)$ with 80% of missing values.

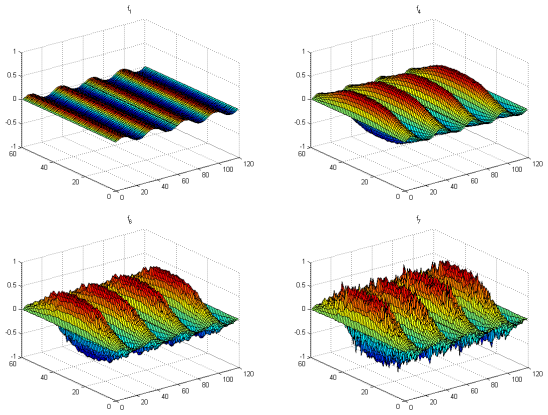

Fig. 5. Top left: X_1, top right: X_4, bottom left: X_6 and bottom right: X_7 - the finest approximation level.

The second example uses a public dataset from the UCI repository [13]. The data consists of expression levels of 77 proteins and has in it many missing values. A subset of complete data X, of size $M \times N = 1000 \times 66$ was taken for evaluation. The results are compared with the WinMice software [7] that implements the multiple imputation algorithm.

Figure 6 plots a $3D$ view of the full data matrix X, where the values are sorted (sorting was applied to simplify visualization). Next, 20% of the data is randomly removed and marked as missing entries. The multi-scale imputation procedure is applied, 8 iterations are needed. The imputed matrixes X_0, X_4, X_6 and X_8 are displayed in Figure 7.

The experiments are repeated with 50% and 80% of missing data. Table I displays the root mean square error for the imputation with the two-directional Lapalcian pyramids and Winmice.

V. Conclusion

In this paper, we presented a multi-scale approach for modeling a dataset with respect to its dual-geometry structures in different scales. The application to data imputation is

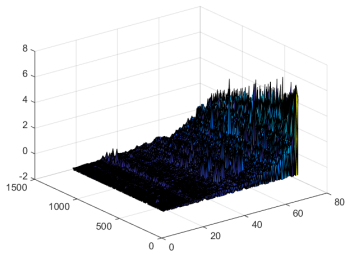

Fig. 6. Protein dataset (sorted)

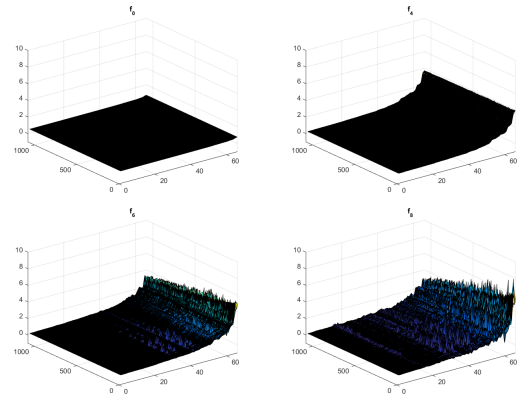

Fig. 7. Top left: X_0, top right: X_4, bottom left: X_6 and bottom right: X_8 - the finest approximation level.

TABLE I
Root mean square imputation errors

% missing	2D-LapPyds ($L^\star$)	WinMice
20%	0.1483	0.2722
50%	0.1564	0.2692
80%	0.2622	0.2787

immediate and the missing data is completed in one pass, together with the model construction. The general representation extended and applied to other learning tasks such learning functions over datasets while considering the two-directional geometric structures.

References

[1] M. T. Asif, N. Mitrovic, L. Garg, J. Dauwels and P. Jaillet, *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] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and Computational Harmonic Analysis, vol. 21, pp. 5–30, 2006.

[3] C. J. Dsilva, R. Talmon, N. Rabin, R. R. Coifman and I. G. Kevrekidis, *Nonlinear intrinsic variables and state reconstruction in multiscale simulations*, The Journal of chemical physics, vol. 139, issue 18, pp. 11B608-1, 2014.

[4] Á. Fernández, N. Rabin, D. Fishelov and J. R. Dorronsoro, *Auto-adaptative laplacian pyramids for high-dimensional data analysis*, arXiv preprint, arXiv:1311.6594, 2014.

[5] Á. Fernández, N. Rabin, D. Fishelov and J. R. Dorronsoro, *Auto-adaptive Laplacian Pyramids*, In: 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, Bruges, Belgium, pp. 59–64, 2016.

[6] M. Huisman, *Missing data in behavioral science research: Investigation of a collection of data sets*, Kwantitatieve Methoden, vol. 57, pp. 69–93, 1998.
[7] G. Jacobusse, *Winmice users manual*, TNO Quality of Life, Leiden, URL http://www. multiple-imputation. com, 2005.
[8] J. A. R. Little and B. D. Rubin, *Statistical Analysis with Missing Data*, 2nd Edition Wiley, 2002.
[9] E. Pierson and C. Yau, *ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis*, Genome Biology, vol. 16, pp. 241, 2015.
[10] N. Rabin and R. R. Coifman, *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.
[11] N. Rabin and D. Fishelov, *Missing Data Completion Using Diffusion Maps and Laplacian Pyramids*, International Conference on Computational Science and Its Applications, pp. 284–297, 2017.
[12] D. B. Rubin, *Multiple imputation after 18+ years*, Journal of the American statistical Association, vol. 91, issue 434, pp. 473-489, 1996.
[13] http://archive.ics.uci.edu/ml/datasets.
[14] Y. UshaRani and P. Sammulal, *An efficient disease prediction and classification using feature reduction based imputation technique*, In: International Conference on Engineering & MIS (ICEMIS), 2016.
[15] D. van Dijk, J. Nainys, R. Sharma, P. Kathail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy and D. Pe'er. *MAGIC, A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data*, Preprint (bioRxiv.org), DOI: 10.1101/111591, 2017