# Ensembles of Multi-scale Kernel Smoothers for Data Imputation

Amit Shreiber
*Dept. of Industrial Engineering*
*Tel-Aviv University, Israel*
Amitamit1@gmail.com

Dalia Fishelov
*Dept. of Mathematics*
*Afeka Tel-Aviv College of Engineering, Israel*
daliaf@afeka.ac.il

Neta Rabin
*Dept. of Industrial Engineering*
*Tel-Aviv University, Israel*
netara@tauex.tau.ac.il

2025 International Joint Conference on Neural Networks (IJCNN) | 979-8-3315-1042-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IJCNN64981.2025.11228219

***Abstract*—When collecting a dataset, it usually contains some proportion of incomplete data. Various methods for handling this missing data exist in the literature, such as deleting observations that contain missing values, or replacing missing values with the mean of the other observations in the relevant variables. Nevertheless, most of the techniques do not consider the geometric structure of the data both in the row (instance) space and the column (feature) space.**

**In this work, we propose a smoothing or regression procedure that operates both on the row and column space of the data, and refines the approximated model in an iterative manner, following ideas from iterative bias reduction models. We provide a mathematical analysis of the method, as well as test its performance of several datasets with diverse missingness mechanisms. Promising results are seen across all of the missingness types and datasets. Last, the proposed multi-scale approximation is general, and may be beneficial for additional machine learning tasks that process tabular data.**

***Index Terms*—multi-scale, kernel regression, imputation, tabular data**

## I. Introduction

Handling missing data, known as data imputation, is a common pre-processing task in data-related applications. When the amounts of missing or corrupted data are large, the use of simple versus sophisticated data completion techniques may determine the quality of the learning model that utilizes the data set. Hence, although classical techniques for data completion are widely accessible to the scientific community, development of new methods is ongoing, with an aim to improve general and specific data completion tasks.

A straightforward approach to managing missing data is deletion, which removes samples with incomplete records. Deletion is easy to understand, simple to implement, and fast to execute, making it the default method in many applications, and a reasonable choice when the proportion of missing data is small [20]. However, this approach has limitations, such as the potential loss of a significant amount of data or the introduction of bias.

Single imputation methods, such as mean imputation and regression, do not remove missing values but instead estimate replacements. Mean imputation assumes that the mean of observed values provides the best estimate for a missing value within a given variable. Its primary advantage is simplicity [2]. However, this approach has limitations, including underestimating the variable's variance, disregarding relationships between variables, and introducing bias in covariance and correlation estimates.

Simple regression imputation, on the other hand, estimates missing values by constructing a regression model in which the variable with missing data serves as the dependent variable, while other relevant observed variables act as independent predictors [3], [4]. Multiple imputation offers notable advantages over single imputation methods by addressing their key limitations. By generating $m$ imputed values, it accounts for the uncertainty associated with missing data. As noted by Bennett (2001) [20], the final dataset reflects the additional variation introduced by missing values. A widely used multiple imputation technique is multivariate imputation by chained equations (MICE) [5]. However, when a dataset contains nonlinear or complex relationships among its variables, the basic regression models used in MICE may struggle to capture these dependencies effectively.

Kernel methods is a widely used approach for capturing relationships between data instances derived from real-world phenomena. Kernels play a crucial role in many unsupervised algorithms designed to find compact representations that reflect the dataset's underlying structure. Additionally, they are fundamental in regression methods, which provide a straightforward way to model the relationship between functions that are defined over scattered data points. In the context of data imputation, kernel-based regression techniques can overcome limitations of linear based regression imputation techniques, which assume that the relationship between the known data points and the target variable to be imputed is linear. Qin et. al. developed a kernel-based missing data imputation method that aims to make an optimal inference on statistical parameters such as the mean and the distribution function [6]. Zhang et. al proposed a a kernel-based stochastic non-parametric multi-imputation method to impute in cases of different missingness mechanisms [7]. A kernel-based method for multiple imputation was proposed in [9] to predict both the missing variable and the probability of missingness. A multi-kernel interpolation approach is suggested in [10] for retrieve missing ratings in the user–item interaction matrix.

Manifold learning and nonlinear dimensionality reduction methods rely on kernels to capture the local geometric structure of the data. This compact representation may be utilized for data imputation. In [8], the data features are assumed

to reside close to a smooth manifold, the tangent spaces to the manifold are then used for regressing the missing values. Dimensionality reduction and clustering was applied for imputation of medical data [11]. The multi-view missing data problem was studied in [13], completion of missing values was done by construction of a multi-manifold regularized non-negative matrix factorization approach.

Recent work propose to take advantage of the geometric structure of both the row and the column space. In [12] a co-clustering technique that solves an optimization problem for filling in the missing values with smooth ones is proposed. Recovery of missing EEG data was explored in [14] by applying nonnegative matrix factorization in a tensor manner. One limitation of kernel-based and manifold based imputation techniques is the global computational nature, which processes the entire dataset to create smooth coordinates that represent the data. Another gap in this line of work is that single imputation methods are typically proposed when utilizing kernels as the main model ingredient. A preliminarily version of this work that suggested a single imputation technique was proposed in [15] and [16]. In [16] a single imputation method, with a different approximation approach was proposed. This work extends the work in [15] by introducing a multiple imputation method, mathematical analysis and experimental results that test several types of missingness mechanisms.

In this paper, we continue the line of work that models both the row and column geometric structures, while taking into account multi-scale relationships. To overcome the mentioned prior limitations, we propose a method that works on many small sub-sets of the data, allowing to generate multiple imputation values for each missing point in aim of improving a global model.

The rest of the paper is organized as follows. Section II outlines that proposed method as a single imputation technique. Extension to multiple imputation is detailed in Section III. Mathematical formulation of our scheme as a general way to approximate a function defined over a tabular dataset, as well as error analysis, are provided in Section IV. Experimental results are described in Section V. Conclusions and future directions are discussed in Section VI.

## II. Multi-scale smoothers for Data Imputation

The proposed method is based on the construction of Nadaraya-Watson regression [17], which successfully models the relationship between a dataset $X$ and the a function $f$, even when the relationship is between them is not nonlinear. The estimator is defines as

$$\hat{f}_\sigma(x) = \frac{\sum_{i=1}^{N} K_\sigma(x - x_i) f_i}{\sum_{i=1}^{N} K_\sigma(x - x_i)},$$

where $K_\sigma(t) = \frac{1}{\sigma} K(\frac{t}{\sigma})$ is a kernel of at least first order $\left(\int_{-\infty}^{\infty} t K_\sigma(t) dt = 0\right)$ with bandwidth $\sigma$.

To refine the regression model, it may be evoked in an iterative manner that reduces the bias. Practically, in the second iteration the residual $f(x) - \hat{f}(x)$ is smoothed, and in the following iterations the process continues, smoothing the modes of the data that have not been approximated in earlier iterations.

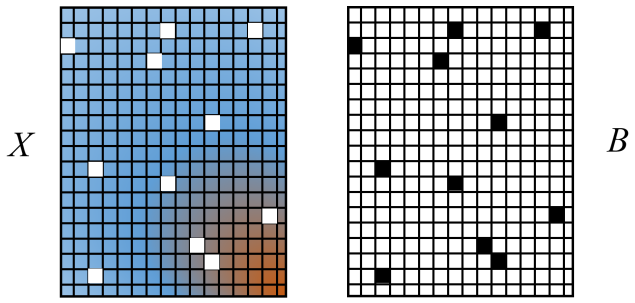


Fig. 1. Illustration of a dataset with missing values (left) and an indicator matrix (right).

In this work we build on the successive Nadaraya-Watson model, but modify the procedure to capture relationships between the rows and columns of a given dataset. When the data has in it missing values, the model is built based on the known entries and imputes the missing values throughout the construction.

Denote the dataset with missing values by $X = (x_{ij})$, a matrix of size $M \times N$. In order to utilize the connections between the rows and columns in $X$, the dataset needs to be normalized. Thus, each column (or each row) should be adjusted such that its mean equals zeros and its variance equals one. Let $B = (b_{ik})$ be a binary indicator matrix of size $M \times N$ that specifies the missing data locations in $X$. Thus, if $b_{ik} = 1$ then $x_{ik}$ contains a known value and if $b_{ik} = 0$ then $x_{ik}$ is a missing data entry. Figure 1 illustrates $X$ and $B$.

Given the dataset $X$ and its corresponding indicator matrix $B = (b_{ik})$, the two initial coarse kernels are constructed based on the known entries of $X$. Here, Gaussian kernels denoted by $G_0^{(L)}$ and $G_0^{(R)}$ are used to define the normalized kernels $K_0^{(L)}$ and $K_0^{(R)}$. The row smoothing kernel that operates on the left is defined by

$$G_0^{(L)} = g_0^{(L)}(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{\sigma_0^{(L)}}}, \quad x_i, x_j \in X,$$

where $x_i$ and $x_j$ are the $i^{th}$ and $j^{th}$ rows of $X$. The distance $\|x_i - x_j\|$ in the exponent is computed as follows.

$$\|x_i - x_j\|^2 = \sum_{\substack{k=1 \\ b_{ik}=b_{jk}=1}}^{M} |x_{ik} - x_{jk}|^2.$$

Therefore, the entries which are included in this distance contains only indices $k$ for which both $x_{ik}$ and $x_{jk}$ are known. This process results in a full matrix $G_0^{(L)}$.

Next, in order to avoid overfitting, a modification is performed on the kernel $G_0^{(L)}$ by setting the diagonal of $G_0^{(L)}$ to zero. Last, the kernel $G_0^{(L)}$ is normalized to be the smoothing operator $K_0^{(L)}$, as each row sum after the normalization is equal to 1.

The same process is applied for construction of $G_0^{(R)}$. Let $x^i$ and $x^j$ be two columns of the matrix $X$, then the elements of $G_0^{(R)}$ are given by

$$G_0^{(R)} = g_0^{(R)}(x^i, x^j) = e^{\frac{-\|x^i - x^j\|^2}{\sigma_0^{(R)}}}, \quad x^i, x^j \in X.$$

The distance $\|x^i - x^j\|$ is then computed by $\|x^{k,i} - x^{k,j}\|$ where $k$ satisfies $b_{ki} = b_{kj} = 1$. Thus,

$$\|x^i - x^j\|^2 = \sum_{\substack{k=1 \\ b_{ki}=b_{kj}=1}}^{N} \|x_{ki} - x_{kj}\|^2.$$

Therefore, the entries which are included in this distance contains only indices $k$ for which both $x_{ki}$ and $x_{kj}$ are known. A normalization of $G_0^{(R)}$ yields the coarse kernel $K_0^{(R)}$. We note that setting a zero diagonal in the left kernel is sufficient for preventing overfitting, there is no need to set a zero-diagonal in the column kernels.

In order to convolve the the data matrix $X$ with the kernels, $X$ needs to be a full matrix. We construct the matrix $X^\star$ by simply imputing the missing values in $X$ with the mean value of the known data entries. Denote this mean value by $m^\star = \text{mean}\,(X_{ij})$, where $ij$ are indices that satisfy $b_{ij} = 1$. Thus, $X^\star$ is constructed as follows

$$X_{ij}^\star = \begin{cases} X_{ij}, & \text{if } b_{ij} = 1 \\ m^\star = \text{mean}\,(X_{ij}), & \text{if } b_{ij} = 0. \end{cases} \tag{1}$$

Then, a coarse approximation of the dataset is computed by

$$X_{\mathbf{0}} = \frac{1}{2}(K_0^{(L)} * X^\star + X^\star * K_0^{(R)}). \tag{2}$$

Eq. (2) holds two terms. $K_0^{(L)} * X^\star$ is a smoothed version of $X^\star$ according to the pairwise connections in the row space, and $X^\star * K_0^{(R)}$ is a smooth version of $X^\star$ according to the pairwise connections in the column space.

The error between the known data values and their coarse approximation from Eq. (2) is computed and stored in $err_{\mathbf{0}}$. It is calculated based on the difference between the original data matrix $X$ and the smoothed matrix $X_{\mathbf{0}}$, based only on known entries for which $b_{ij} = 1$, denoted by

$$err_{\mathbf{0}} = \|X_{ij} - X_{\mathbf{0}\,ij}\|.$$

Here $err_{\mathbf{0}}$ is computed as the root mean square error.

The first residual is given by $D_{\mathbf{1}} = X - X_{\mathbf{0}}$. The values of $D_{\mathbf{1}}$ are known for all locations $ij$ that satisfy $b_{ij} = 1$. The matrix $D_{\mathbf{1}}^\star$ is then constructed as described in Eq. 1, the missing values in $D_{\mathbf{1}}$ are replaced by the mean of its known entries. The kernels $K_1^{(L)}$ and $K_1^{(R)}$, which operate on $D_{\mathbf{1}}^\star$ from the left and from the right, yield a more accurate representation of $X$, denoted by $X_{\mathbf{1}}$. It is given by

$$X_{\mathbf{1}} = X_{\mathbf{0}} + \frac{1}{2}(K_1^{(L)} * D_{\mathbf{1}}^\star + D_{\mathbf{1}}^\star * K_1^{(R)}).$$

The construction is carried out in an iterative manner and the iterations continue for a pre-defined maximal number of steps. In the $l-th$ iteration the residual $D_l = X - X_{l-1}$ is computed, $D_l^\star$ is constructed. A finer representation of $X$, which is given by

$$X_l = X_{l-1} + \frac{1}{2}(K_l^{(L)} * D_l^\star + D_l^\star * K_l^{(R)}).$$

The error, denoted by $err_l$, is calculated based on the difference between $X$ and $X_l$ on the known data entries. Since we set a zero-diaginal in the left smoothing kernels, this process will not overfit the data, and after several iterations that refine the approximation of $X$, the errors will begin to grow. Thus, the process stops in the iteration the results with the minimal error values.

The proposed approach is summarized in Algorithm 1.

**Algorithm 1:** Imputation with two-directional Laplacian pyramids

**Input:**

- Dataset $X$ of size $M \times N$, normalized, with missing values.
- A location indicator matrix $B$ of size $M \times N$.
- $\sigma_0^{(L)}$, $\sigma_0^{(R)}$ - initial kernel widths.
- $l_{max}$ - maximum number of iterations.

**Output:**

- Multi-scale imputed representation of $X$: $\{X_0, X_1, \cdots, X_l\}$.
- The values in $X_{l\,ih}$ for which $b_{ij} = 0$ are the imputed data for the missing values in $X$.

Construct $K_0^{(L)}$ and $K_0^{(R)}$
Construct $X^\star$.
$X_{\mathbf{0}} = \frac{1}{2}(K_0^{(L)} * X^\star + X^\star * K_0^{(R)})$.
Compute the root mean square error $err_0$ and store it in $err[0] = err_0$
**for** l=1 *to* $l_{max}$ **do**
    $D_l = X - X_{l-1}$.
    $D_l^\star$.
    $X_l = X_{l-1} + 0.5(K_l^{(L)} * D_l^\star + D_l^\star * K_l^{(R)})$.
    $err[l] = err_l$
**end for**
Determine the scale $l$ for which $err[l]$ reaches its minimum value.
**return** $\{X_0, X_1, \cdots, X_l\}$, where $X_l$ is the final result.

The smoothing procedure that is described in Alg. 1 sets a similar weight for the row and column kernels. This can be modified into a more general setting, in which one can control the contribution of the row and column kernels to the sum. In this setting $X_l$ is expresses by

$$X_l = X_{l-1} + \left(\alpha\left(K_l^{(L)} * D_l\right) + (1-\alpha)\left(D_l * K_l^{(R)}\right)\right),$$

where $0 \leq \alpha \leq 1$. In this work, $\alpha$ is set using a grid search.

To illustrate the main building blocks of Alg. 1, we plot in Figure 2 the left and right multiscale kernels of the Housing dataset that is described in the results section.

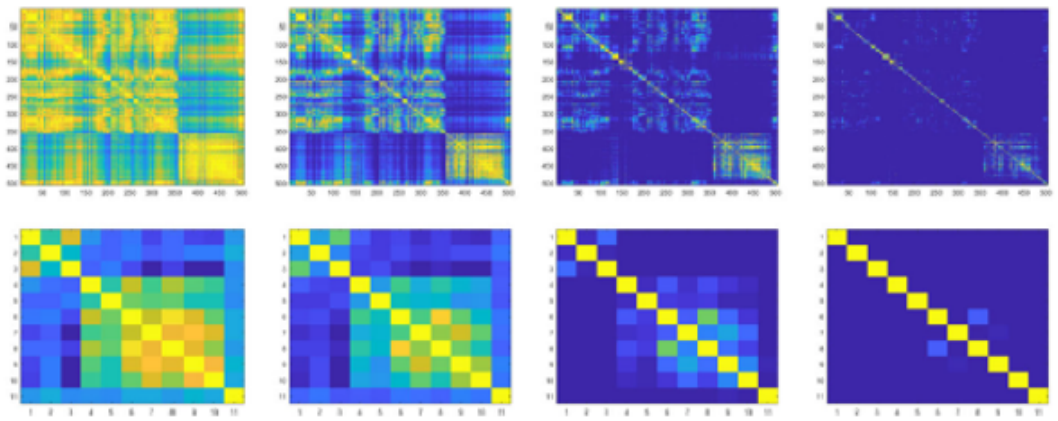

Fig. 2. Top: Multiscale left smoothing kernels $K_0^{(L)}$ - $K_3^{(L)}$ that capture the connections in the row-space of the Housing dataset in different resolutions. Bottom: Multiscale right smoothing kernels $K_0^{(C)}$ - $K_3^{(C)}$ that capture the relationship between the 11 features (columns) of the Housing dataset. (Figure reproduced from [16])

### A. Parameter Setting

The initial kernel widths $\sigma_0^{(L)}$ and $\sigma_0^{(R)}$ can be determined by estimating the pairwise distances—between the rows of $X$ for the left kernel and between the columns of $X$ for the right kernel. Here, the following MaxMin heuristic (see [18] ) is used

$$\sigma_0^{(L)} = 2 \cdot \max_j [\min_{i,i\neq j} (|x_i - x_j|)^2],$$

and

$$\sigma_0^{(C)} = 2 \cdot \max_j [\min_{i,i\neq j} (|x^i - x^j|)^2].$$

The parameter $l_{max}$ that appears in the for-loop of Alg. 1 determines the maximal number of imputation-approximations to the dataset. In this work $l_{max}$ was set to 10. We note that since the left (row-based) kernels are constructed with a 0-diagonal, there is no risk of data overfiiting. In practice, the errors, $err_l = ||X_{ij} - X_{l\ ij}||$ decrease for several iterations and then begin to increase (this happens when the kernels scales become to fine and overfit the data). The errors are kept in the array $err[l]$ (see Alg. 1) and the number of iterations is set in accordance to the lowest error. From our experimental settings, typically, only a small number of iterations is required for reaching the optimal number of iterations $l$, thus setting $l_{max} = 10$ is sufficient. To reduce computational complexity, one may replace the for-loop with an until-loop and stop the iterations when $err[l] < err[l+1]$.

## III. Ensembles of Multi-scale Kernel Smoothers

In this section, we describe how to improve the performance of Algorithm 1 to work in a multiple imputation setting instead as a single imputation method. The overall scheme is based on sampling the original dataset into smaller sub-parts, then applying Algorithm 1 to each sampled dataset. Figure 3 illustrates this idea.

If the missing value is imputed in more than one subset, the algorithm leverages the multiple predicted values for imputation and determines a single value for the missing data point. Two sampling techniques are proposed and compared. The first method, denoted by *Shuffling and Sampling*, uses the hypergeometric distribution to determine the validity of a sampled subset; the second is a Naïve approach that shuffles and splits the data in subsets of smaller size. We denote is as *Naïve Sampling*.

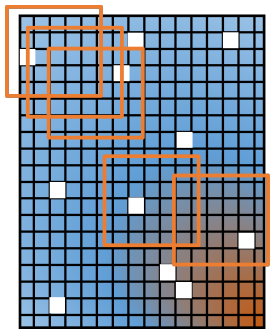

Fig. 3. Illustration of our proposed ensemble imputation technique. Alg. 1 is applied on subsets of the data. Missing values may reside in several subsets, resulting with multiple imputation values that are avaraged.

In the Shuffling and Sampling approach, a subset is considered *valid* if it contains at least $k_{\min}$ data points with missing values. The probability that a subset of known size contains at least $k$ instances of relevant missing points is calculated using the Hypergeometric distribution. $k_{\min}$ is the minimal value that ensures that this probability is smaller than a pre-defined value $p_{\max}$.

## IV. Mathematical Formulation and Error Analysis

The proposed scheme is a relaxation process that in the continuous case interpolates the data. In order to gain insight on the error decay rate and size, we analyze the behavior of the process in terms of a function approximation method. Like before, $X$ is a dataset of size $M \times N$, and we assume that

$f = f(x, y)$ is a function that is defined on $X$. In our case $f = X$, but for simplicity of notations we carry on with $f$ in this section.

Assume that $f$ is in $L_2$, i.e., $\int_x f^2(x)dx \leq K$, for some constant $K$. Define the kernels $k_l^{(L)}(x)$ and $k_l^{(R)}(x)$ which approximate a delta function. The kernel $k_l^{(L)}$ operates on $f$ from the left and $k_l^{(R)}$ operates on $f$ from the right. The two kernels satisfy

$$\begin{array}{l} \int k_l(x)\,dx = 1, \\ \int x k_l(x)\,dx = 0, \\ \int |x|^2 |k_l(x)|\,dx \leq C. \end{array} \tag{3}$$

Note that $k_l^{(L)}$ and $k_l^{(R)}$ are normalized kernels. In this work we choose

$$\begin{aligned} k_l^{(L)} &= c_l^{(L)} e^{-x^2/\sigma_l^{(L)}}, \qquad \sigma_l^{(L)} = \sigma_0^{(L)}/\mu^l \\ k_l^{(R)} &= c_l^{(R)} e^{-x^2/\sigma_l^{(R)}}, \qquad \sigma_l^{(R)} = \sigma_0^{(R)}/\mu^l, \end{aligned} \tag{4}$$

where $c_l^{(L)}$ and $c_l^{(R)}$ are normalizing factor for $k_l^{(L)}$ and $k_l^{(R)}$ respectively.

The two-sided scheme is a relaxation process for which in the first step the function $f$ is approximated by

$$f_0 = \frac{1}{2}\left(\tilde{K}_0^{(L)} * f + f * \tilde{K}_0^{(R)}\right)$$

Define

$$d_1 = f - f_0,$$

then, in the second step $f$ is approximated by

$$f_1 = f_0 + \frac{1}{2}\left(\tilde{K}_0^{(L)} * d1 + d1 * \tilde{K}_0^{(R)}\right)$$

Taking the Fourier transform of $k_l^{(L)}(x)$ and using the assumptions in Eq. (3) for $k_l^{(L)}(x)$, we have

$$\begin{aligned} &\left|\hat{k}_1^{(L)}(\omega) - 1\right| \leq C(\sigma_l^{(L)})^2 \|\omega\|_2^2, \text{ where} \\ &C = \tfrac{1}{2}\int_{-\infty}^{\infty} x^2 |k_1(x)|\,dx. \end{aligned} \tag{5}$$

Similarly for $k_1^{(R)}$.

We first analyze the error in the first step. The error $d_1(x)$ is defined by

$$d_1(x) = f(x) - \frac{1}{2}\left(\tilde{K}_0^{(L)} * f + f * \tilde{K}_0^{(R)}\right).$$

Taking the Fourier transform of $d_1(x)$ and using Equation (5) we have

$$\begin{aligned} \left|\hat{d}_1(\omega)\right| &= \tfrac{1}{2}\left|\hat{k}_0^{(L)}\hat{f}(\omega) - \hat{f}(\omega) + \hat{k}_0^{(R)}\hat{f}(\omega) - \hat{f}(\omega)\right| \\ &\leq \tfrac{1}{2}\left|\hat{k}_0^{(L)}\hat{f} - \hat{f}(\omega)\right| + \tfrac{1}{2}\left|\hat{f}(\omega)\hat{k}_0^{(R)} - \hat{f}(\omega)\right| \\ &= \left|(\hat{k}_0^{(L)} - 1)\hat{f}(\omega)\right| + \left|\hat{f}(\omega)(\hat{k}_0^{(R)} - 1)\right|. \end{aligned} \tag{6}$$

First, note that by Taylor expansion of $\left|\hat{k}_0^{(R)}(\omega)\right|$ around $\omega = 0$, and by using Equation (3) for $k_l^{(R)}(x)$, it follows that is bounded $\left|\hat{k}_0^{(R)}\right|$ is bounded by a constant. Bounding the two terms on the right-hand-side of (6), we have

$$\left|\hat{k}_0^{(R)}\right|\left|\hat{f}(\omega)\right| \left|(\hat{k}_0^{(L)} - 1)\hat{f}(\omega)\right| \leq C(\sigma_0^{(L)})^2 \|\omega\|_2^2 \left|\hat{f}(\omega)\right| \tag{7}$$

and

$$\left|\hat{f}(\omega)(\hat{k}_0^{(R)} - 1)\right| \leq C(\sigma_0^{(R)})^2 \|\omega\|_2^2 \left|\hat{f}(\omega)\right|. \tag{8}$$

Here $C$ denotes a universal constant.

Combining Equations (6), (8) and (7), we have

$$\left|\hat{d}_1(\omega)\right| \leq C((\sigma_0^{(R)})^2 + (\sigma_0^{(L)})^2)\|\omega\|_2^2 \left|\hat{f}(\omega)\right|. \tag{9}$$

For simplicity we assume that $\sigma_0^{(L)}$ and $\sigma_0^{(R)}$ are bounded by $\sigma_0$. Thus,

$$\left|\hat{d}_1(\omega)\right| \leq C\sigma_0^2 \|\omega\|_2^2 \left|\hat{f}(\omega)\right|. \tag{10}$$

The error in the second step is

$$d_2 = d_1 - \frac{1}{2}\left(\tilde{K}_0^{(L)} * d_1 + d_1 * \tilde{K}_0^{(R)}\right) \tag{11}$$

Taking the Fourier transform of Equation (11) yields

$$\left|\hat{d}_2(\omega)\right| = \left|\hat{k}_1^{(L)}\hat{d}_1(\omega)(\omega)\hat{k}_1^{(R)} - \hat{d}_1(\omega)\right|. \tag{12}$$

It may be bounded as in Equation (10) by

$$\begin{aligned} \left|\hat{d}_2(\omega)\right| &\leq C((\sigma_1^{(R)})^2 + (\sigma_1^{(L)})^2)\|\omega\|_2^2 \left|\hat{d}_1(\omega)\right| \\ &\leq C(\sigma_0)^2(\sigma_0/\mu)^2\|\omega\|_2^4|\hat{f}(\omega)|. \end{aligned} \tag{13}$$

For the $l^{th}$ level the error is bounded by

$$\left|\hat{d}_l(\omega)\right| \leq C\sigma_0^2\left(\frac{\sigma_0^2}{\mu^l}\right)^{l-1}\|\omega\|_2^{2l}\left|\hat{f}(\omega)\right|. \tag{14}$$

By Parseval's equality we obtain

$$\|d_l(x)\|_{L^2} \leq C\sigma_0^2\left(\frac{\sigma_0^2}{\mu^l}\right)^{l-1}\|f(x)\|_{2l,2}. \tag{15}$$

Thus, the error for the two-sided smoothing procedure as well decays faster than any algebraic rate.

## V. Experimental Results

Experimental results evaluate the proposed multiple imputation approach on four public datasets from the UCI repository. The datasets are Ecoly, Housing, Wine Quality and Frogs. Table I describes their properties.

TABLE I
DESCRIPTION OF THE DATASETS

| Data Set | Num. of rows | Num. of Columns |
|---|---|---|
| Ecoly | 366 | 7 |
| Housing | 506 | 13 |
| Wine | 4,898 | 11 |
| Frogs | 7,195 | 21 |

In addition, we consider three missingness mechanism (as classified by Rubin, 1976 [19]): missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR). Missing at random (MAR) occurs when

the probability of missing data for a specific variable in a dataset does not depend on the values of that variable itself, but on the values of other observed variables in the dataset. Thus, the pattern of missingness is traceable or predictable from other observed variables in the dataset [20]. Missing completely at random (MCAR) is a private case of MAR that occurs when the probability of missingness does not depend on both observed and unobserved data [21]. This effectively implies that the causes of the missing data are unrelated to the data. If neither MCAR nor MAR holds, the data is missing not at random (MNAR). MNAR means that the probability that a variable value is missing depends on the missing data values themselves [22].

The above three missingness mechanisms are considered in the results, which are evaluated in terms of Mean Squared Error (MSE). The percentage of missing values in each dataset is noted in each experiment. We compared our results with several imputation techniques including mean substitution (mean), most frequent value substitution (freq), Scikit-learn iterative imputer, inspired by the Multiple Imputation by Chained Equations (MICE) [25], and the proposed method in a single imputation setting, without the ensembles. We denote our single imputation technique (Alg. 1) by *Single MSSI*, where MSS stands for Multi-Scale Smoothing Imputation. Out multi-imputation versions are denoted by *MSSI-SS* and *MSSI-NS*, where MSSI-SS denoted the Shuffling and Sampling approach and MSSI-NS uses the Naive shuffling approach.

### A. MAR Imputation Results

The MAR generation procedure follows the mechanism described by Santos et al. (2019) [24]. For each variable chosen to be unobserved, the method considers two variables: the variable chosen to include missing values (the unobserved variable), and another observed variable that determines the unobserved variable missingness pattern. The method includes the following steps. First, the user chooses the number of missing variables and the threshold percentile cut-off. This combination determines the dataset's total number of missing values, so it should be selected carefully. This step forms a pair of unobserved and observed variables. The unobserved variable is chosen by drawing one variable from all the variables in the dataset. Then, the most correlated variable (different from the drawn one) is chosen as the corresponding observed variable. This procedure is repeated until the desired number of missing variables is reached. Each time the unobserved variable must be a new variable that was not already chosen to be observed or unobserved in the previous pairs. The corresponding observed variable cannot be a variable that was chosen to be unobserved in the current or prior iterations. Finally, values in the observed variables that exceed the chosen threshold percentile cut-off are identified for each pair of variables. The corresponding values in the unobserved variables are replaced with missing values. Table II displays the results, with the lowest errors marked in bold. It can be seen that except for the Wine dataset, our proposed method performs well.

TABLE II
MAR IMPUTATION RESULTS (MSE)

| Dataset (% missing) | Mean | Freq. | MICE | Single MSSI | MSSI SS | MSSI NS |
|---|---|---|---|---|---|---|
| Ecoly (5.7%) | 1.48 | 2.88 | 0.47 | 0.38 | **0.31** | 0.41 |
| Housing (5.38%) | 1.47 | 3.36 | 3.04 | 1.23 | **1.03** | 1.06 |
| Wine (5.24%) | 1.12 | 1.44 | **0.79** | 0.82 | 0.84 | 0.84 |
| Frog (20%) | 1.06 | 35.93 | 0.95 | 0.62 | 0.6 | **0.59** |

### B. MCAR Imputation Results

MCAR generation is straightforward and intuitive. Random elements in the original observed dataset are replaced with missing values to create an unobserved MCAR dataset. The proportion of total missing values is determined by a parameter that the user sets. Table III plots the results. It can be seen that our proposed method has an advantage when processing datasets with this type of missingness. This is due to the connections both in the row and column space as well as the iterative refinement.

TABLE III
MCAR IMPUTATION RESULTS (MSE)

| Dataset (% missing) | Mean | Freq. | MICE | Single MSSI | MSSI SS | MSSI NS |
|---|---|---|---|---|---|---|
| Ecoly (5%) | 0.66 | 0.63 | 1.42 | 0.34 | **0.33** | 0.41 |
| Housing (5%) | 1.04 | 1.75 | 0.55 | 0.39 | 0.39 | **0.38** |
| Wine (5%) | 0.98 | 1.25 | 0.61 | **0.48** | 0.49 | **0.48** |
| Frog (20%) | 0.99 | 21.65 | 0.23 | **0.19** | 0.2 | 0.2 |

### C. MNAR Imputation Results

MANR generation involves selecting a proportion of unobserved variables in the new dataset [23]. For example, if the original dataset has 100 columns and a proportion of 20% is chosen, then 20 columns in the new dataset will be designated as unobserved variables. Next, a random percentile cut-off value between 30% and 60% is drawn for each column, and all values below this cut-off percentile in the relevant column are replaced with missing values. For example, if the cut-off point for a variable is 50%, all values lower than the median are removed. Results are presented in Table IV. It is noticeable that the MICE method struggles with this type of missingness pattern, while our proposed method performs well.

TABLE IV
MNAR IMPUTATION RESULTS (MSE)

| Dataset (% missing) | Mean | Freq. | MICE | Single MSSI | MSSI SS | MSSI NS |
|---|---|---|---|---|---|---|
| Ecoly (27.2%) | 1.48 | 1.39 | 4.21 | 1.26 | **1.25** | 1.33 |
| Housing (22.4%) | 1.18 | 1.86 | 1.25 | **0.9** | 0.91 | 0.92 |
| Wine (15.5%) | 1.26 | 1.22 | 1.26 | 1.17 | **1.07** | **1.07** |
| Frog (20.5%) | 0.83 | 99.67 | 4.83 | 0.88 | **0.82** | **0.82** |

## VI. CONCLUSIONS

This work proposes a multi-scale smoothing approach for modeling and imputing missing data in a dataset. The proposed

method is inspired by the multiple imputation approach as well the iterative regression. Its strength lies in considering the geometric structure of the row and column spaces, using kernels of decreasing widths. We provide error analysis of the method, suggesting that also in real data applications we expect the error to decrease fast. Another advantage of the method is that it does not suffer of convergence risks, and will always result with an approximated version of the data. The iterative nature of the approximation bypasses the need to find an optimal single bandwidth for the smoothing kernels, and also, as a byproduct of the construction derives a multi-scale imputed representation of the dataset.

Two methods were introduced and examined for creating sub-datasets. Naïve Shuffling that splits the dataset in a naive way to equal parts and Shuffling and sampling, which creates subsets that include sufficient missing points, determined using the hypergeometric distribution.

To evaluate the effectiveness of the proposed methods, experiments were conducted on several datasets with different dimensions and three different missingness mechanisms: missing at random (MAR), missing completely at random (MCAR), and not missing at random (MNAR). The mean squared error of the methods' imputed values was calculated and compared to other benchmark impute methods. The proposed method show stable and low error results across all setting, typically with a slight advantage to the ensemble approach.

Even though recent research by Le Morvan and Varoquaux [26] suggest that accurate imputation methods may play a minor role when strong predictive models are used, we believe that the suggested construction may be beneficial for other machine learning tasks. For example, the proposed approached may be applied to detect anomalies in tabular datasets, based on the difference between the smooth approximation of the data and the original data. Moreover, it may be considered for subset selection of large training sets, where the goal is to identify complementary subsets that characterize the set both in the instance and feature space.

## VII. Acknowledgments

This research was supported by the Israel Science Foundation [Grant 1144/20].

## References

[1] D. A. Bennett, "How can I deal with missing data in my study?," *Aust N Z J Public Health*, vol. 25, no. 5, pp. 464–469, 2001.

[2] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *J. Big Data*, vol. 8, pp. 1–37, 2021.

[3] C. M. Musil, C. B. Warner, P. K. Yobas, and S. L. Jones, "A comparison of imputation techniques for handling missing data," *West. J. Nurs. Res.*, vol. 24, no. 7, pp. 815–829, 2002.

[4] D. J. Stekhoven and P. Bühlmann, "MissForest—non-parametric missing value imputation for mixed-type data", *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.

[5] S. Van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *J. Stat. Softw.*, vol. 45, pp. 1–67, 2011.

[6] Y. Qin, S. Zhang, X. Zhu, J. Zhang, and C. Zhang, "POP algorithm: Kernel-based imputation to treat missing values in knowledge discovery from databases," *Expert Syst Appl.*, vol. 36, no. 2, pp. 2794–2804, 2009.

[7] S. Zhang, Z. Jin, X. Zhu, and J. Zhang, "Missing data analysis: A kernel-based multi-imputation approach," *Trans. Comput. Sci. III*, pp. 122–142, 2009.

[8] D. T. Nguyen and K. Slavakis, "Multilinear kernel regression and imputation via manifold learning", *IEEE Open J. Signal Process*, vol. 5, pp. 1073-1088, 2024.

[9] C.-H. Hsu, Y. He, Y. Li, Q. Long, and R. Friese, "Doubly robust multiple imputation using kernel-based techniques", *Biom. J.*, vol. 58, no. 3, pp. 588–606, 2016.

[10] Z. Chen, W. Zhao, and S. Wang, "Kernel meets recommender systems: A multi-kernel interpolation for matrix completion", *Expert Syst Appl.*, vol. 168, pp. 114436, 2021.

[11] Y. UshaRani and P. Sammulal, "An efficient disease prediction and classification using feature reduction based imputation technique,", *Proc. 2016 Int. Conf. Eng. & MIS (ICEMIS)*, pp. 1–5, 2016.

[12] M. Zhang, G. Mishne, and E. C. Chi, "Multi-scale affinities with missing data: Estimation and applications", *Stat. Anal. Data Min*, vol. 15, no. 3, pp. 303–313, 2022.

[13] J.-T. Sun and Q.-Y. Zhang, "Completion of multiview missing data based on multi-manifold regularised non-negative matrix factorisation", *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5411–5428, 2020.

[14] M. Akmal, S. Zubair, and H. Alquhayz, "Classification analysis of tensor-based recovered missing EEG data", *IEEE Access*, vol. 9, pp. 41745–41756, 2021.

[15] N. Rabin, "Multi-directional Laplacian pyramids for completion of missing data entries", *Proc. ESANN*, pp. 709–714, 2020.

[16] N. Rabin and D. Fishelov, "Two directional Laplacian pyramids with application to data imputation", *Adv. Comput. Math.*, vol. 45, no. 4, pp. 2123–2146, 2019.

[17] E. A. Nadaraya, "On estimating regression", *heory Probab. Appl.*, vol. 9, no. 1, pp. 141–142, 1964.

[18] N. Rabin, M. Golan, G. Singer, and D. Kleper, "Modeling and analysis of students' performance trajectories using diffusion maps and kernel two-sample tests",' *Eng. Appl. Artif. Intell.*, vol. 85, pp. 492–503, 2019.

[19] D. B. Rubin, "Inference and missing data", *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[20] D. A. Bennett, "How can I deal with missing data in my study?", *Aust. N. Z. J. Public Health*, vol. 25, no. 5, pp. 464–469, 2001.

[21] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art", *Psychol. Methods*, vol. 7, no. 2, pp. 147, 2002.

[22] D. A. Newman, "Missing data: Five practical guidelines", *Organ. Res. Methods*, vol. 17, no. 4, pp. 372–411, 2014.

[23] R. Wei, J. Wang, M. Su, E. Jia, S. Chen, T. Chen, and Y. Ni, "Missing value imputation approach for mass spectrometry-based metabolomics data", *Sci. Rep.*, vol. 8, no. 1, pp. 663, 2018.

[24] M. S. Santos, R. C. Pereira, A. F. Costa, J. P. Soares, J. Santos, and P. H. Abreu, "Generating synthetic missing data: A review by missing mechanism", *IEEE Access*, vol. 7, pp. 11651–11667, 2019.

[25] M. J .Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?", *Int. J. Methods Psychiatr. Res.*, vol. 20, no. 1, pp. 40–49, 2011.

[26] M. L. Morvan and G. Varoquaux, "Imputation for prediction: beware of diminishing returns," in *Proc. 13th Int. Conf. Learn. Represent. (ICLR)*, 2025.